Table of Contents

Chapter 1:
The roadmap to Product/Market Fit (PMF)... maybe

PERSONAL FIT · MARKET FIT · CUSTOMER FIT
SHIP THE SLC · MARKETING & SALES · RETENTION
PRIORITIZATION · PSYCHOLOGY
IGNORING IT ANYWAY · STORY OF SMART BEAR
STORY OF WP ENGINE

"What was initially thought to be a simple process is in fact an incredibly complicated, intricate, and complex system that I've codified and organized into a few easy-to-follow rules that are more difficult to implement than you'd think."

credit[1]

What is the formula for going from an initial product idea to Product/Market Fit—a company that is growing and sustainable, with customers that want to pay and want to stay.

(Assuming you even agree with that definition of Product/Market Fit (p. 335).)

There isn't a fail-safe roadmap of course, but there is a progression that describes how WP Engine became a unicorn, supported by 18 years of articles in this book.

Here's what that progression looks like. Then we'll explore the other truth—that it often doesn't work like this after all.

1. **Personal Fit**: "Passion" is useful, but winning requires a personal edge.
2. **Market Fit**: Most good ideas aren't good businesses.
3. **Customer Fit**: Talk to customers before you waste months building the wrong thing.
4. **Build and ship the SLC** quickly: It's where the real learning happens.
5. **Marketing and Sales** more than writing code and tweaking design.
6. **Retention-driven Product** development; attend to your existing customers first.
7. **Prioritize** systematically, ruthlessly, strategically.
8. **Manage your psychology** on a journey of self-discovery and constant rejection.

1. PERSONAL FIT: THE INNER FIRE, LEVERAGING A PERSONAL EDGE TO DO WHAT YOU WERE MEANT TO DO

A great idea or a great strategy that *you* can't execute well, isn't a good strategy *for you*.

You start with nothing—no product, no customers, no brand, no distribution, and compared to competitors, no money and no time.

To succeed, you need *something*. Something special. Something that gives you an edge, despite being woefully inadequate in every dimension. Is "passion" that something?

Passion

> *"Do what you love, and the money will follow."*
>
> *–Motivational advisors*

That is false, as evidenced by most artists, philosophy majors, and the 80% of startups that fail despite founders' genuine love and obsession.

Passion *is* required; it's just not enough. Passion is the motivator, especially in the dark times when your savings is depleted, competi-

tors are beating you, as you suffer a constant bombardment of rejection from potential customers and potential employees and potential investors, as you question your self-worth (p. 737) and wonder whether the cause of your Impostor Syndrome (p. 457) is that you're actually an impostor (p. 449).

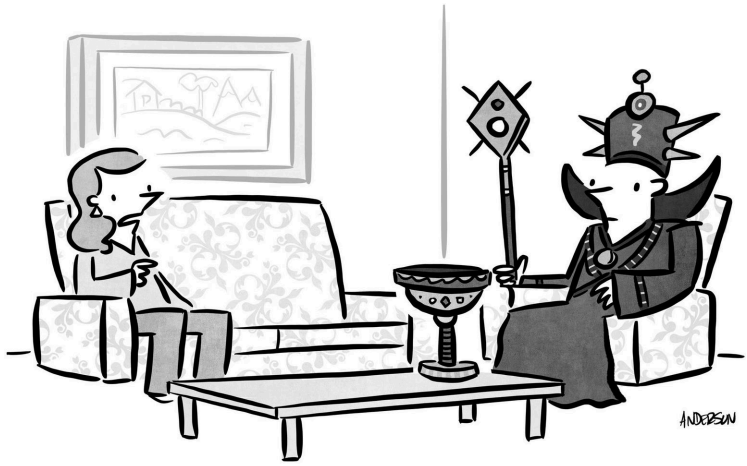
> *"You have to be burning with an idea, or a problem, or a wrong that you want to right. If you're not passionate enough from the start, you'll never stick it out."*
>
> *—Steve Jobs*

You need to articulate exactly what your passion is, so that you can use it as a filter for business concept. This is difficult; some people spend their whole life trying to figure it out. Jerry Colonna says[2] "The purpose of life is to discover who you are, and then live fully into that." It is not an exaggeration to say that a startup is who you are (p. 1005); that's why they call it "your baby."

See this article on Pivot Points (p. 569) for a specific list of questions that help you suss what your purpose and passions are.

Your goal is to **find the work you were *meant* to do**. Your calling. The work that you would do for free, but more—what you are *compelled* to do.

Still, the existence of passion doesn't imply the existence of a business model. (We'll solve that in the next step.) Passion doesn't give you a competitive edge, because all the other founders have passion, and larger companies have mountains of advantages that make "passion" look like a sling-shot attacking an aircraft carrier. You need an edge.



'Listen, I'm fine with the robe, the sorcery, even mixing potions in the bathtub. But for god's sake, Gary, would you *please* use a coaster?!"

credit[3]

Leverage

"Leverage" means yielding a huge output from a given input—accomplishing an inordinate amount of quantity or quality, with relatively little time and money. You generate leverage from the mixture of talents, taste, and experiences that you possess. To figure out how this works for you, see this article for many types and examples of leverage (p. 543).

Leverage is good, but *unique* leverage is far better, because that's your source of differentiation* from competitors and alternatives. Often this appears at the intersection of your peculiar above-average talents and experiences, which taken together are unique.** This

* "Different," not "better." Yes, you will be "better" according to some set of people, but definitionally that means you're "worse" for another set. The latter might even be orders of magnitude more numerous than the former. That's OK, and that's why I don't like using the word "better." You should be distinctive, aligned with your calling and your strengths, so that the set of people who do find your brand of different to be "better" will flock to you, buy from you, and even love paying you (p. 275) for it.

uniqueness must then be coupled with a particular way that you are approaching this problem and solution, such that the combination of your leverage against your path is uniquely excellent, even for a small number of potential customers (p. 317).

This leverage + path combination also forms the kernel of your strategy*** as your competitive positioning. You need something that competitors lack, because your customers are going to compare you to competitors whether you like it or not, and you have to have an answer besides "we have this one minor feature that they don't have" or "we're $10 cheaper."

The key question that summarizes this intersection of "you" and "path" is:

Why are *you* the perfect person to build *this* company?

> *"Be yourself. Everyone else is taken."*
> *—Oscar Wilde*

** The quintessential exponent of this idea is Scott Adams, writer of the Dilbert cartoon, who points out[4] that he is a decent illustrator but not a great one, a decent humorist but worse than any comedian on Netflix, and has held jobs in the tech sector, unlike almost any artist or comedian. It is in the intersection of being in the top 25% in each of these three "circles of competence" that he is unique, not because he is top 1% or "best in the world" at any one thing.

*** Leveraging strengths into durable competitive advantage is one of the six characteristics of great strategy (p. 489).

2. MARKET FIT: A WORKING THEORY OF WHY THEY WILL BUY.

Once you inventory the inner world—the shapes of spaces where you were meant to play—you turn your gaze externally to figure out whether your idea works out in the market, in the world of customers, competitors, trends, problems-to-solve, jobs-to-be-done, and products. These things you do not control, and thus must understand, conform to, but also exploit.

It's right there in the name: "Product/Market Fit" means fitting into the market, not just building something that would be fun/interesting/edifying/curious/exciting/ego-enhancing. Those are all good reasons to have a side-project, but none is a reason why that side-project will become a profitable business, even at a scale that feeds a single person. That's why most "indie hacker" startups and "AI is cool" startups fail—the genesis was "fun project," not "plausible business."

"I had the problem myself, so I built a product to solve it."

This might be the most common origin story, tacitly concluding that "this must be a business because I would have been a customer." Indeed, my startups all started this way. But your understanding of "the problem" as it pertains to you alone is much less likely to be a real business than you think. You are, in fact, not like your customer.*

Sometimes a passion project turns into a business anyway. That's what happened to me at Smart Bear. More on that later. But that's luck. The point of a framework like this one is to reduce your reliance on luck.

> "The way to get startup ideas is not to try to think of startup ideas. It's to look for problems, preferably problems you have yourself.

* For a start, your customer isn't quitting their day job to create a business.

> The very best startup ideas tend to have three things in common: they're something the founders themselves want, that they themselves can build, and that few others realize are worth doing. Microsoft, Apple, Yahoo, Google, and Facebook all began this way."
> *–Paul Graham, How to Get Startup Ideas*[5]

A theory of why they will buy

You need a plausible theory of the customer, market, and business model. "Plausible," because many ideas fall apart under honest scrutiny. "Theory" because of course it won't be entirely correct, but it will be your working understanding of the world, which you modify as you learn.

Here's exactly how to develop a specific theory of the market, customer, and your positioning **(p. 71)**. You'll analyze these market characteristics:

1. **Plausible:** Do 10M people or 100k companies have the problem?
2. **Self-Aware:** Do they know & care they have the problem?
3. **Lucrative:** Do they have substantial budget to solve this problem?
4. **Liquid:** Are they willing and able to buy right now?
5. **Eager:** Do they want to buy from you, specifically?
6. **Enduring:** Will they still be paying(-it-forward) a year from now?

A lot of people read that article, then said[6] they[7] wish[8] they[9] had[10] read[11] it before[12] they[13] wasted two years of their life (and savings and heartache) building [insert name of failed startup]. Don't be one of those people.

You might think it infeasible to answer these questions, because the research is impossible or the idea is so new that existing data and trends are irrelevant. However, as you'll see in the article, you *can* do it using Fermi Estimation—a technique useful not only in market analysis but for ROI (p. 171), probabilities (p. 997), and decisions (p. 603).

credit[14]

"Well, it's not the worst I've seen."

Early strategy

Even better than having a great business model is to have a great strategy (p. 489). Not a twenty-page document, but a one-pager that conforms to the guidelines in that article, explaining "how we will win." It's never too early to be asking yourself how to leverage your strengths (p. 543) to build products that are great and differentiated despite your weaknesses (p. 891). It's never too early to write down your assumptions and decisions, to ensure they are at least self-consistent if not self-reinforcing, so that when contrary evidence appears, you can notice that, and react methodically.

To generate tangible strategic ideas for your business model or strategy—explaining "how we will win"—consider creating a theory of where your product will fit in the customer's Needs Stack (p. 259), pick a few of these tactics for navigating the fact that the future is unpredictable (p. 193), and decide how you will generate "Love" and "Utility" types of Willingness-to-Pay (p. 275). All of these are fun, accessible, strategic, and they work in practice.

"But often the first idea is wrong, so why do I need a specific idea with a specific business model?"

Because walking in a random direction is not progress, and because the main way to discover the *right* direction, is to have a clear theory of the world, and notice when the world contradicts it, so you can pivot into a better theory, and thus iterate into a genuinely great strategy and market-understanding.

That's what *actually* happened with Smart Bear, and Slack and WhatsApp and Flickr, as described and analyzed here (p. 193). Only by having a specific, strong, clear idea, was it possible to notice what customers *really* wanted *instead*, which led each of those companies to pivot into the idea that became successful.

In all those cases, the idea was very personal to the founders—the personal-fit of passion for the solution and leveraging an edge to create something that resonated with early users. But it was only with a specific theory of business model and strategy, which was then negated by reality, and then an intentional pivot, to achieve Product/Market Fit.

3. CUSTOMER FIT: FIND THE IDEAL CUSTOMERS FIRST.

There's only one source of truth for what customers will buy: Customers.

Not advisors, not experts, not analysts, not Twitter polls, not research, not past data,[15] not even competitors' behavior. **You have to talk to customers.**

Customers are fickle. You ask whether they'll pay $100 if your product does _____ and they say "yes;" then you build it and they

"So, tell me a little bit of what you think I want to hear about yourself."

credit[16]

don't buy. So why talk to them at all? Don't you need to put the actual product in front of them, and see what they actually do?

Customers can tell you what their lives are like, which is how you validate your business model and strategy from the previous section. Customers can tell you what they *won't* buy, which has happened to me repeatedly. You can discover that the average customer doesn't know the problem exists, or doesn't have a budget for it, or isn't prioritizing it. When they say "yes," it's a "maybe," but when they say "no," it's a "no," and you just saved yourself months or years of wasted time.

You can find out where they go to find products like this, so that you can advertise in the right place. You can find out what language they use to talk about the problem or solution, so you can copy that language in your advertising and social media home page and capture their attention. You can find out how their budgets work, so you can price and package and position accordingly. You can find out how they're addressing the problem today, so you know what you're selling against, whether that's a competitor, an alternative, or something they're doing by hand. You can find out what causes them to break

out of their daily life and say as Bob Moesta puts it,[17] "Today is the day I'm going to buy _____," so that you can try to be there when that event happens, or possibly even cause it.

It is easy to find examples of successful companies who never asked customers what they wanted. That happened to me at Smart Bear. But again that relies on luck, and most of the time, you don't get lucky. This is not a good way to gamble the next few years of your life.

But you don't want to do this work. You want to build the product because that's the fun part. Potential customers are hard to find and they don't want to talk to you. It's going to be like that when you have a product too, so if you can't do it now, you won't be able to do it later. Building the product first won't make it easier to find or talk to customers. Building the product first will, however, ensure that you haven't actually built what people want, because you never found out what people want.

How to interview customers

Once you get someone on the phone, how do you interview them in a way that maximizes learning and leads to a specific theory that you can execute on, from your unique winning advantage[19] to pricing (p. 165)?

Justin ✓
@JustinQuda

It's taken years for me to change my mindset from "I know best" to "what does the user want?"

I've been building a startup in my own echo chamber.

Today I did an 1 hr call with a Quda creator - it was awesome!

Talk to users - they won't bite. They want you to succeed too.

credit[18]

The customer validation system I've developed is the Iterative-Hypothesis customer development method **(p. 239).** This provides you with goals that you achieve using interviews, how to create the hypotheses that will drive your business model, and how to write questions that maximize learning. 16 years ago it invalidated a startup idea that I *thought* was good, and then validated the startup idea that became WP Engine—now a unicorn.

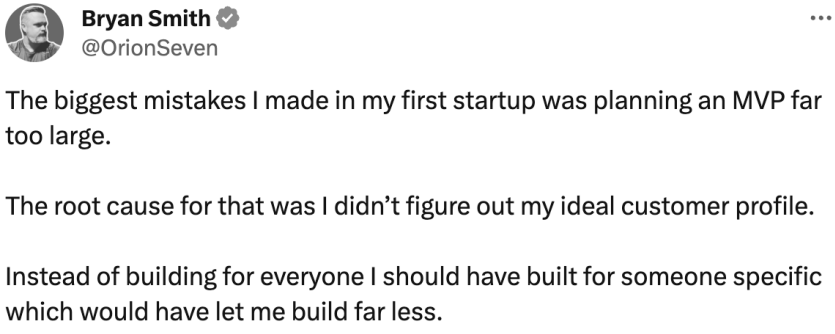
How to get customers to talk to you

The way I got interviews for WP Engine is by using LinkedIn to find people who had the title and industry I was targeting (web developers in WordPress), and asked them for an hour of their time to chat about a new startup concept for whom they are the ideal customer. Furthermore I offered to pay any amount they wished for that hour, because I value their expertise and their time; I'm not asking for a donation, I'm genuinely interested in their expertise. Out of 50 requests, 40 agreed, and only one asked to be paid. (30 eventually became customers.)

This might not work for you; that was a long time ago. **This article details many more techniques for finding potential customers** **(p. 683)**, by going where they already are, where they're already talking, and where they might be willing to talk to you.

Find your ICP

Your goal is not only to validate your theory of the market, but to discover your ICP (Ideal Customer Profile) (p. 317). This is your "perfect customer"—a segment almost comically over-specified to be so perfect for your product that you are truly the best choice in the market, and they would be crazy not to buy. You will then aim all of your marketing messaging at this person: Website, advertisement, terminology, writing style, pricing. When the ICP lands on the home



credit[20]

page, it should be obvious in three seconds that this is perfect for them, that you "get" them, and they will be pulled through the process until they're a customer.

You are scared that targeting only the ICP limits your potential market, but this is not what happens. In "Selling to Carol (p. 317)," I explain the mechanism and provide examples showing that for every ICP there are 10x more people who make similar buying decisions, and 100x more who take more convincing but ultimately also agree, and therefore you end up selling to a far wider market than you feared, yet earning attention and loyalty from having a clear, unique message. Your company, brand, and product will actually mean something.

Find and talk to the customers; they're the only ones with the answers.

4. BUILD *AND SHIP* THE SLC.

Interviews give you a better model of the world, but the real learning doesn't start until people are using—or more often, *not* using—a product.

credit[21]

"I'm looking for something in a small pond."

SLC is my alternative to MVP (p. 101):

Simple, because complex things can't be built quickly, and you must ship quickly so you can learn quickly so you can create the right product before you run out of money and willpower.

Lovable, because crappy products are insulting, and you didn't start this company to make crappy products. The love overpowers the fact that the product is buggy and feature-poor. There are many wonderful, powerful, competitively-defensible forms of "Love." (p. 275) Pick some.

Complete, because products are supposed to accomplish a job. Customers want to use a v1 of something simple, not v0.1 of something broken.

credit[22]

Build something small (but also lovable and complete), and ship it to customers, because that's when real "progress" begins. That's how you learn the truth (p. 657).

If you're building for six months and a customer hasn't used it, you're not on the road to Product/Market Fit; indeed you're not making progress on the business at all. What you've done is proved what you already knew—you can build software.

The learning starts only when customers are (trying to) use it.

5. FOCUS MORE ON MARKETING AND SALES, LESS ON PRODUCT.

Your natural inclination is to do what you enjoy and understand, so you need to press yourself to do so much marketing and sales that it feels like "too much."

Ask yourself: What can you do *today*, that will (a) get more people from the target market to come to the website or (b) convert more of those people to try the product or (c) convert more of those to buy the product.

The answer *might* be found in writing more code. Improving the new-user experience, for example, could increase the conversion rate from "try" to "buy." But more often the answer isn't inside the product; indeed several of those steps happen before the person has used the product at all. The code is actually your enemy. (p. 635)

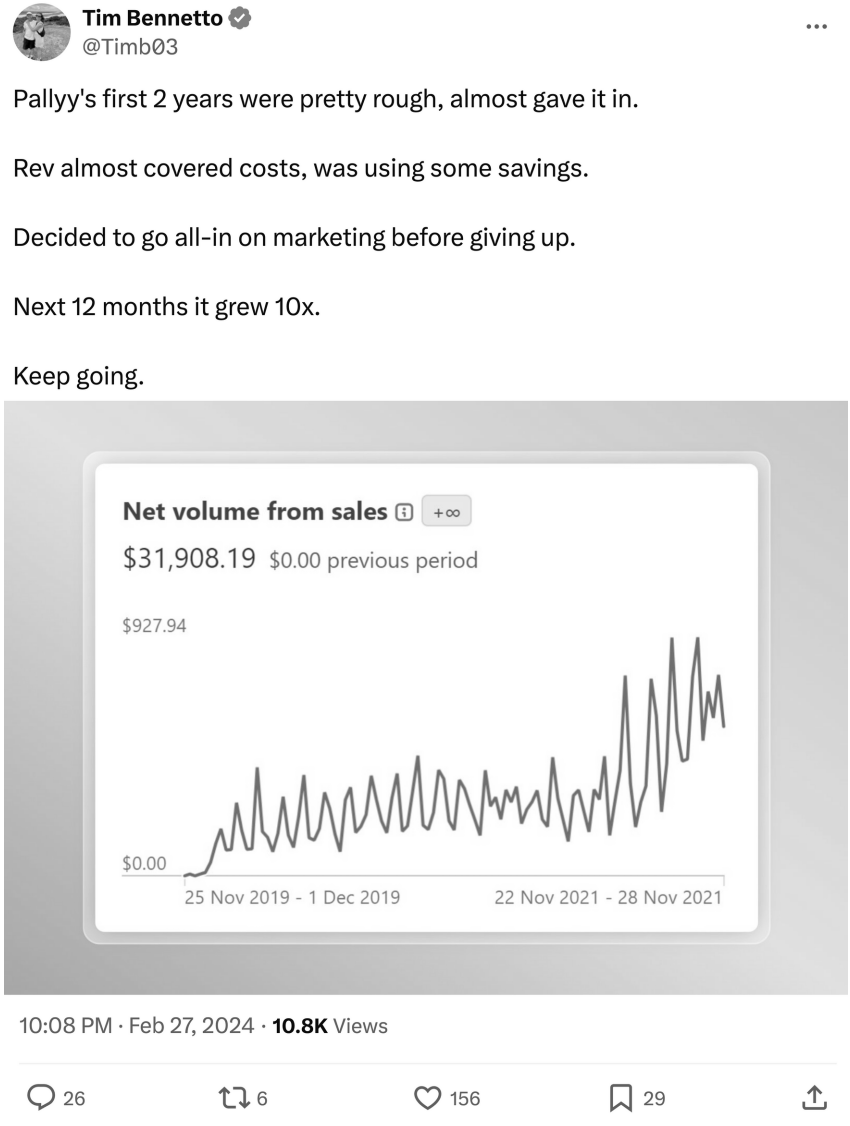
You'll advance the product anyway—I know that. And you should, but you don't need to be told to do that. After all, it's all you ever wanted to do, even at step one. That's why you need to set an intention every day to make progress on one of those questions. That's how you're going to generate growth.

6. PRIORITIZE RETENTION-BASED PRODUCT WORK.

What does it mean, that one person out of billions cared enough about your problem-space to notice your advertisement or social media post, then felt compelled to follow the link, then was so intrigued by the website that they joined the 1-out-of-100 that tried the product, then liked what they saw enough to take out their wallet and pay, then started to use it... and then *cancelled?!?*

After all that—clear signals of desire, need, willingness to pay (p. 275), and apparent "fit"—they decided "Nope, this is not what I want."

What it means is: You made the right promise, but didn't keep it.



credit[23]

Marketing is where you discover the promise the customer wants you to make, and retention* is where you discover whether you're fulfilling that promise.

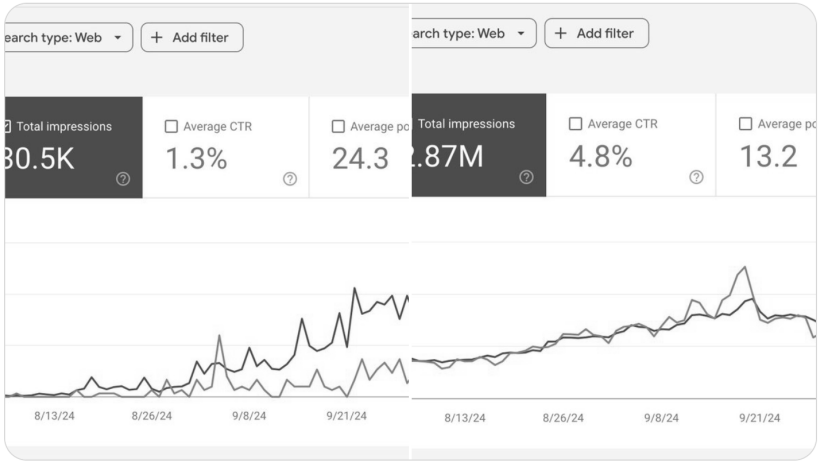
Stefan Wirth
@NafetsWirth

I barely worked on code in the last 3 months.

Instead, I've actually made money.

By focusing on growth.

You should try it out.



credit24

Some people believe that low retention is fine because you can make up for it with growth. That's not true for two reasons:

1. Growth will start out good, but then will slow and halt sooner than you realize.**
2. If customers don't want to use the product, your "growth" is fake news. It's not Product/Market "Fit" if the "market" decides "not a fit" after actually using the product.

---

* Actually, retention is a lagging, multi-factor indicator, which makes it useful as a high-level check, but not useful as an operational day-to-day metric. Here's a complete system for practical metrics (p. 645) that embraces this fact and points to the right set of metrics.

credit[25]

"No, you're thinking of Cupid. I'm the one-night-stand fairy."

Therefore, when you do work on product features, build things that increase retention.

This starts by understanding why people cancel. Gaining this understanding is difficult, because once they leave they don't want to talk to you anymore. You might coerce them by paying for interviews with Amazon gift cards or charitable donations. Some people will respond to an email (no harm in asking). Better is to look for signs that the customer isn't being successful (e.g. activity within the product) and reach out before it's too late, while they're still in the middle of their struggle, and therefore might be willing and able to articulate it.

Remember that your goal is not to discover what unsuccessful customers have in common, because successful customers often have those things in common as well. Rather, it is to find patterns in unsuccessful customers *that are not shared by successful customers*. Those are the attributes that lead to action.

** See these articles with data from Product/Market fit companies (p. 335) and the math behind growth grinding to a halt (p. 71) at companies with low retention.

For example, if "using feature X" means people are successful, maybe unsuccessful customers don't know about it (so you promote its usage), or maybe they can't use it because it's incompatible with their workflow (so add integrations or options). Sometimes, however, it's about the customers themselves, where tuning the ICP means you find more customers who are already a fit, perhaps targeting different industries or geographies, different stages or sizes of company, the title or attitude of the buyer, or the specific use-cases they are trying to solve for.

The Talk/Walk workshop (p. 949) might help, because it helps elucidate the difference between the promise and the execution. But probably, as usual, you just need to talk to customers.

7. RUTHLESSLY, METHODICALLY, SYSTEMATICALLY PRIORITIZE.

More than just "focus," more than a generic admonition to "prioritize," use a system that ensures you're leveraging your most valuable, most limited asset (time) to maximize progress.

Do this by combining Fairytale Planning (p. 1065) with the Rocks, Pebbles, Sand (p. 221) work-prioritization system:

Identify the next milestone.
At all times, be crystal clear about the single most important thing to achieve.

(Ex: Finding a good business model. Public launch. 20 paying customers. Get-

ting trial conversion rates from 1% to 5%. Getting cancellation from 7%/mo to 4%/mo. Getting to $10k MRR so you can quit your day job.)

You have to be executing against the next milestone every day. Write it down and look at it every morning. Everything else is a distraction to be ignored, or a necessary evil that you should dispatch as quickly as possible, including delegation (p. 981) or accepting that it's fine if it's executed poorly or late.

Identify the current obstacle to achieving the milestone.
At all times, be crystal clear why it is difficult or slow or expensive to achieve that milestone.

(Ex: Can't figure out how to leverage your strengths for something differentiated and desirable. Easy to get someone to use it once, but only 15% use it again. Got 17 customers from Product Hunt but now there's no repeatable way to get more. Can't get people to agree to be interviewed. SLC isn't accepted by beta testers. Freemium users don't have enough incentive to convert to paid.)

You must face the difficult truth (p. 657). Do not pick the obstacle that feels comfortable; pick the existential crisis that is uncomfortable exactly because it is critical and scary and true.

You have to be attacking or side-stepping the obstacle every day. Write it down and look at it every morning.

Split work into Rocks, Pebbles, Sand.
Use this specific Rocks, Pebbles, Sand framework (p. 221) to prioritize and schedule different types of work with the correct philosophy and process appropriate for each type.

Select the Rock that attacks the obstacle and milestone together.
Use Binstack (p. 603) to prioritize one Rock that addresses that most important milestone and attacks the obstacle. (You don't have time for more than one.)

If you can't think of one that's good enough, don't just proceed with a mediocre plan that will occupy the next three months; grab some friends—or better yet, customers!—and brainstorm (p. 53) a better one.

Schedule Pebbles sparingly.
Because you have to focus on the Rock, and you have little time for anything else, use this ROI framework[26] to prioritize just a few other activities that are valuable enough to justify spending your remaining time.

Almost everything should be only "good enough"
You have taste, you have craft, and part of the reason you're doing all this is to express yourself. But almost everything you do will not determine whether the company is successful. (Repeat that last sentence until you fully internalize it.) Which means most things should only be "good enough." Let it go.

Most metrics should be satisfied, not maximized (p. 887). Pick KPIs accordingly (p. 645). Design matters less than you think (p. 853).

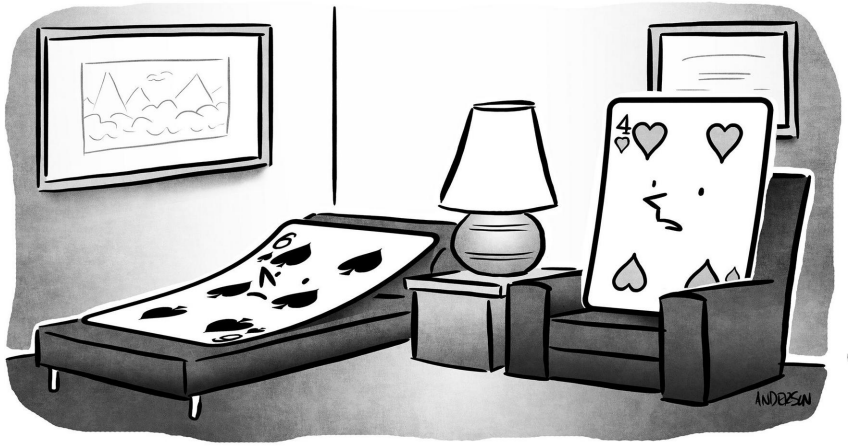
Having said all that, sometimes the very best aspects of a design are the little things, the obsessive things, the things you are compelled to make just so. Whether web design or UX design or a "perfect" feature which is only perfect because of the myriad corner cases that took a surprising amount of code and care to cover, but which means a new customer has an amazing experience. Pick your battles.

Don't stray from the system. You don't have the time.

8. MANAGE YOUR PSYCHOLOGY.

Throughout this a step-by-step roadmap, you face demanding psychological challenges.

- **Facing the truth** **(p. 657)**—not allowing "what you wish were true" to get in the way of finding out what is true, about your abilities, customer's wishes, the shape of the market, why people leave. Recognizing that there's always someone who is better than



credit[27]

"So you'll never be part of a royal flush. You know what? Me neither. And that's OK."

you at any given thing, and it becomes your job to find and hire them (p. 981).

- **Finding yourself**—coming face-to-face with who you are, and who you are not, figuring out what is actually important (p. 827) versus what is necessary or temporary, enjoying the journey (p. 861) especially when even a successful end result (p. 45) is often not what you hoped it would be (p. 1005).
- **Making clear decisions**—deciding what you're not good at, and not trying to do it anyway; deciding your place in the market, and not trying to be all things to all people; deciding your ICP (p. 317) and having the fortitude to go all-in; deciding the current milestone, the main challenge, the one big thing you have to do, and sticking to it without getting distracted.
- **Constant rejection**—your ideas won't be good enough, your customer interviews will negate key assumptions, 99% of website visitors won't take an action, long sales calls will result in nothing, customers will buy but then leave, employees won't join you, investors won't invest.

- **Fluid change**—despite clear decisions, rejections often demand a change, which is difficult to accept, difficult to enact (p. 1299), difficult to admit when things are wrong, and difficult to decide whether some rejection should be ignored as an anomaly, or whether it's indicative of a "learning" that must result in a change, difficult to tell the difference between the chaos that results in PMF and the one that results in failure (p. 429).
- **Acting while uncertain**—being "all-in" even while knowing you must be wrong about some things, even while getting rejected and making changes, even while doing things you've never done while experts tell you what to do differently (p. 751), even while feeling like an impostor who has no right to be here (p. 457), even while knowing you have no idea what you are doing (p. 737) while it appears everyone else knows exactly what they're doing. (Don't worry, they really don't.)

This might be the hardest thing you've ever done. It's a gauntlet of pain,[28] even if it's also the most exciting thing you'll do in your professional life. Many people stop because it's so hard. You can't blame them.

BUT YOU'RE NOT GOING TO DO THIS, ARE YOU…

You have a fun feature idea so you're just building it.

You always wanted to learn Rust so you're just learning it.

AI is astounding and it's the future so you're building things with it, even though it's a technology and not a customer's problem that needs solving.

You've started your own company in part because you don't want to be shackled by "processes" and "frameworks," so you're just doing whatever you want.

Visual Studio is comfortable and Google Ad Manager is foreign and like Las Vegas is designed to separate you from your money, to say nothing of the discomfort of asking for the sale (p. 737) or getting someone on a call to talk about their workflows or get an earful about how your precious software[29] actually sucks.

Uncomfortable. Scary. And you're lost, unskilled at those things, not making progress, not even *wanting* to. So you slip back to Visual Studio where you know what to do, and enjoy doing it, and you're good at it.

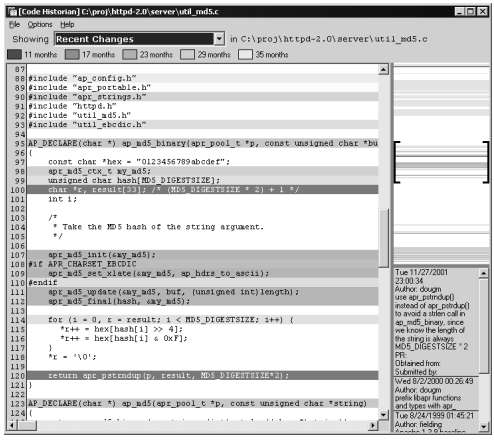
Then you post on Twitter about how you have $1600 in MRR after 18 months but it's okay because perseverance is a virtue. When in fact you've fallen into the well-known trap of doing what you're good at, what you love, but not what the business needs done (p. 399).

Perseverance *is* a virtue, if you're doing the right work, with the right goals. It's a vice if it means you're moving diligently in the wrong direction.

SMART BEAR ONLY HALFWAY FOLLOWED THE ROADMAP

But who am I to talk?

The company I founded in 2002, sold in 2007, and that gave me the online handle that persists to this day on this site, Smart Bear's first "product" Code Historian was a personal project, built because I want



This doesn't make me look old, right?
Right???

to learn .NET (we would later rewrite everything in Java), built because I thought there was value in the historical record encoded in version control (all products that operated under that assumption failed).

I didn't talk to customers ahead of time. I did talk to them a lot after they started paying, and learned that many didn't really want Code Historian at all, but rather they were abusing it to accomplish something else (in a story told here (p. 193)). As a result, I created Code Reviewer, which sort of hit the need in question, but the architecture was all wrong and the features were all hacks. But all this customer feedback gave me the confidence to rewrite everything from scratch (which everyone says you should never do), resulting in a product called Code Collaborator, which made millions of dollars.

Would I have found the idea for Code Collaborator by interviewing customers using the "Iterative-Hypothesis" method above? No, because I wouldn't have known what to ask them, and they wouldn't have known to tell me. I would have discovered that Code Historian wasn't a good business idea... and that's it. Maybe I would have had a completely different and better idea. Maybe. But the fact is, the actual "path to success" was to create something I liked, just interesting

enough to attract initial customers even though it's not a great business, and then pivot to Code Reviewer, and then pivot to Code Collaborator. The interviews would have stopped everything cold. Does that make "interviewing" the wrong road to Product/Market Fit?

I do think building and getting product into customers hands is far more valuable than conversations. Smart Bear was an example of (1) having a specific idea matching my passions, which then because of (4) building and shipping and (5) marketing and selling and (6) building for retention by talking to customers, lead to the right idea. So, it was part of the path, but not the full path.

So, while the path to success didn't honor every step in the roadmap, it still would have failed if I had violated those other steps—coding in isolation, coding instead of selling, coding instead of honestly figuring out what customers really wanted.

WP ENGINE FOLLOWED THE ROADMAP AND BECAME A UNICORN

That said, with WP Engine I traveled this roadmap exactly.

WP Engine wasn't the initial idea. The first idea was for a marketing analytics tool with features that Google Analytics lacked: A real-time event dashboard, a clever goal-setting-and-measuring system that worked retroactively, and the ability to capture web-form data even when it was only partially filled-in and never submitted, all with just one line of Javascript.* But I ran that Iterative-Hypothesis method and after 20 interviews I learned that the idea wasn't strong enough.** I was on the road; I (1) had a specific idea that (2) was clearly a great

* These features are typical of analytics tools today, but in 2009 it was innovative.

market and (3) vetted it with customers. Thanks to (3), I did not waste years on that project.

I then had the idea of a managed WordPress platform because I needed it myself—my blog would crash when I got to the top of Hackernews (a link-sharing site). I asked other bloggers what they use to keep WordPress fast and scalable; many said, "I don't know, but if you find it, tell me, because I need that too." I ran the Iterative-Hypothesis method (p. 239) again, and this time it was very clear what the need was, what people would pay, and how to find them. And the market was already huge (WordPress powered 11% of the largest 10 million websites) yet growing fast (in 2023 it's 43%). I knew WordPress because I was a user, and for nearly twenty years I've loved code optimization problems, so this was a problem I understood and am ideal to solve, where customers agree the problem exists, are already spending money on it, and the market is large and growing. That's steps (1), (2), and (3), but this time the concept was validated.

I built just enough software to fulfill the promises of speed and scale, building the first version in 30 hours. Good enough to not only stay alive and fast on a Hackernews day, but with almost no load on the server, which meant I was still far away from a capacity limit (p. 945) (Figure 1).

The first version didn't even have a customer dashboard. I took credit cards with the default web-form supplied by the online credit card service, and the rest was just WordPress.

The first version of the website was... "spare," shall we say? (Figure 2)

The second version of the website was so bad, there were database errors shooting out the bottom. Database errors, on the home page, of the company that you're supposed to trust with your precious website (Figure 3).

So that's step (4), the SLC product that fulfills the job completely with the absolute minimum of "Product."

** This article gives the details of this invalidation (p. 845), and how it was different from WP Engine, which the same method validated.

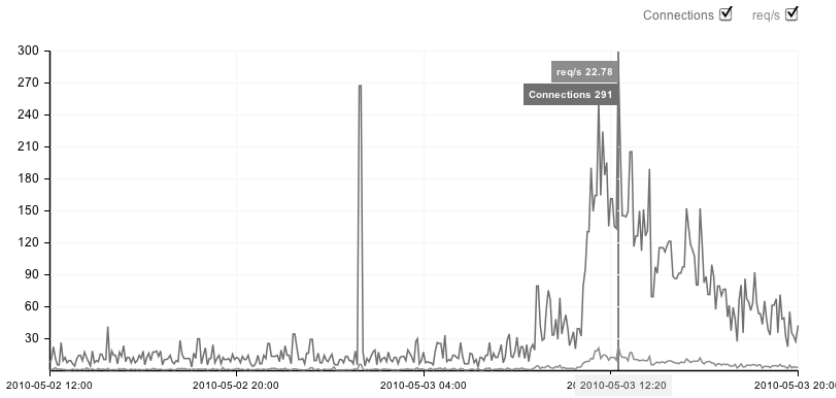


Figure 1: Hundreds of simultaneous connections on a Hackernews day in 2010

Figure 2

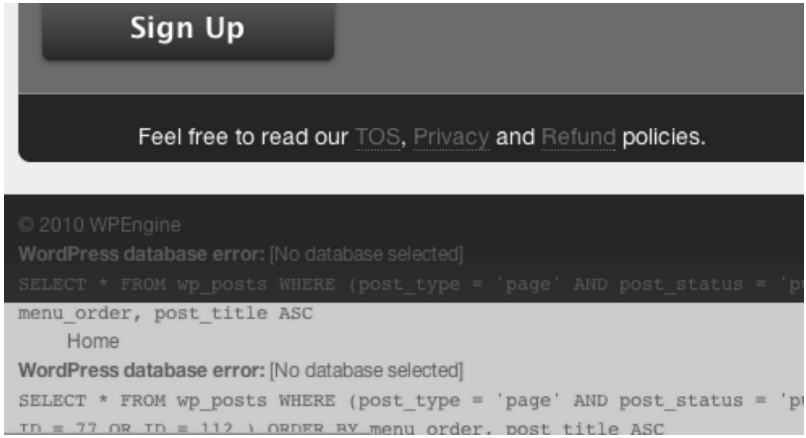


Figure 3: The market need must be overwhelming, if customers walk through this wall of fire and say "yeah just take my money."

QUESTION FOR YOU: Are there places we should be advertising? Sponsoring? Pushing our affiliate program?

QUESTION FOR YOU: Do you know other blogs (on any platform) that should be using us? Please make an introduction!

Figure 4: From an "advisory group" email update from 2010; my asks were for marketing and sales, not product and engineering.

We reached $1M ARR after 18 months. The reason—besides having the right product with the right promise in the right market, which was elements (2) and (3)—is step (5) focus on marketing and sales.

Because I did marketing and sales before even having a product login screen, I was able to launch with 30 paying customers. Sure I worked on the product and worked support tickets, but what I focused on was how to get customers (Figure 4).

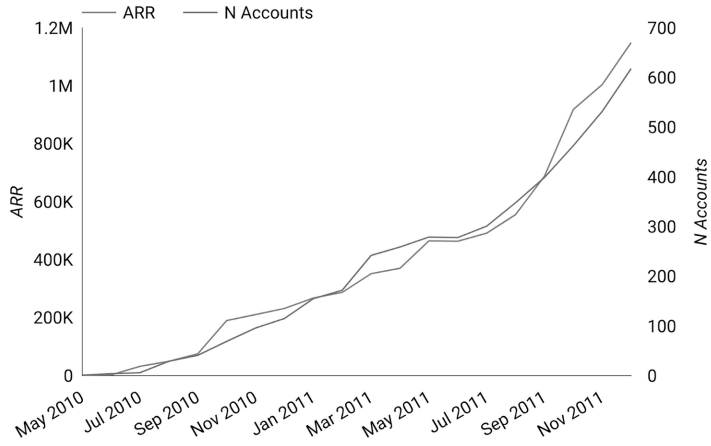


Figure 5: Hit $1M ARR in first 18 months at WP Engine. But wait, the Product/Market Fit is still to come…

When you do marketing and sales, with a product that people actually need, in a large and growing market (all covered by the exercise under (2)), then you might have a nice growth curve (Figure 5).

We've always spent a lot of time on (6) retention. Sometimes years would pass where we wouldn't add a new feature, because all our engineering time was spent on the challenges of scale (p. 773) and continuing to improve on the promise of speed and uptime. As a result, we've had best-in-industry customer retention for 15 years.

High retention due to happy customers folds back into growth through word-of-mouth referrals and a thousand top-of-class reviews (Figure 6).

All that, combined with the hard work, the craftsmanship, the large and growing market—did I mention the large and growing market?—and yes also the luck, meant that in early 2012 we hit Product/Market Fit, as further detailed here (p. 335) (Figure 7).

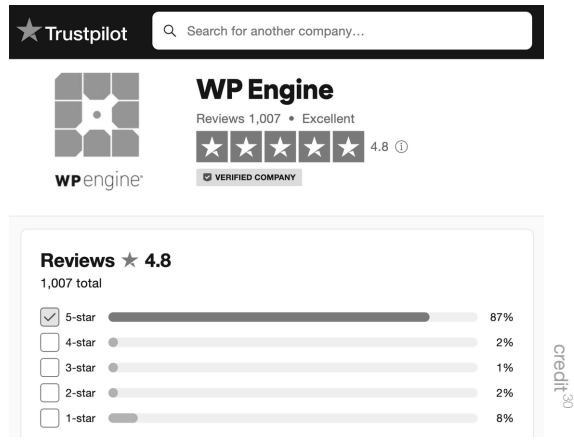


Figure 6

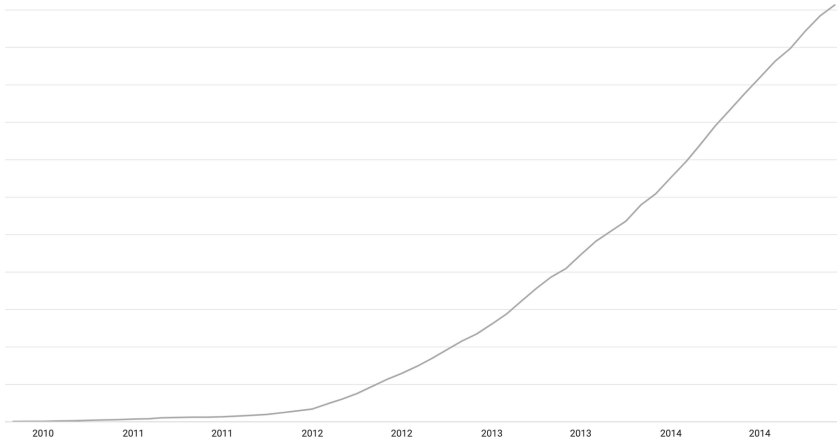


Figure 7: The previous chart's "up and to the right" growth from 2010-2012 looks almost flat compared to the sudden moment of Product/Market Fit, when it became hyper-growth, and never slowed.

This roadmap worked for WP Engine, and it worked for others. Consider this thread from Simon Hoiberg,[31] explaining why his second bootstrapped company succeeded (FeedHive[32]) where his first failed. This roadmap echos within it—playing in a large, growing market, finding and focusing on an ICP, building only an SLC to start, and treating marketing and sales as more important than features and innovation (Figure 8).

Do you *have* to walk this road like Simon discovered with FeedHive and I discovered at WP Engine, or can you stumble onto it halfway through like Smart Bear, or can you take a completely different route and just emerge at the finish line? Or can you just cherry-pick the things you like from this framework, and ignore the rest?

Yes to all the above. Everything is possible.

But not all roads are equally likely to result in success. We enjoy stories about outliers because exceptions are interesting. But exceptions are not repeatable, and therefore they may not be teaching you anything useful. I believe the roadmap above *is* repeatable. Not a guarantee of success, but a process that makes sense.

It's your life; you can take any road you please.

What are you going to do?

Many thanks to Gordon Daugherty,[34] Rowan Udell,[35] and Sathyanand S[36] for contributing insights to early drafts.

simonhoiberg 1w ···

My first startup never made any money.

My second startup passed $10K MRR in year 1, and is doing +$1M ARR today, 3 years later.

What did I do differently?
Let me share the lessons learned 👇🏻

4 replies · 80 likes

simonhoiberg 1w ···

→ I talked to users and listened instead of telling myself that I knew better.

→ I built in a very well-established market with verified demand instead of trying to innovate.

→ I consistently marketed my product - slow, but sustainably - instead of trying to "blow up" using the latest growth hacks.

2 replies · 21 likes

simonhoiberg 1w ···

→ I kept my product slim but effective instead of trying to solve everything for everyone. No one likes a "feature creature".

→ I put UI/UX on a pedestal and had a huge focus on *how* we solve a problem, instead of just solving it "somehow".

→ I focused on users who truly appreciated the unique way we're solving problems differently instead of trying to target everyone.

16 likes

credit 33

Figure 8

Chapter 2:
Rich vs. King in the Real World: Why I sold my company

I sold my company, Smart Bear,[38] in December of 2007. I haven't talked about it at all on this blog, and it's time I spill my guts about the whole affair.

credit[37]

You'd think selling a company would be a glamorous, exuberant experience, but I was surprised at the reactions I got. These are actual quotes:

- "How could you sell your baby? I'm shocked."
- "I thought you said things were going well. Hmm."
- "**You're such a sell-out!** You used to be one of the few cool people I knew."

Interestingly, 100% of the negative reactions were from people who had never started their own company. But that doesn't make them wrong, and it doesn't make their words sting less, especially when they're your friends.

Now that almost two years have passed, I can relate exactly why "selling my baby" was right for me.

Hopefully this thought process is interesting to you and possibly useful in the happy event that you're faced with the same choice, but the truth is I just need to get this off my chest.

I need to explain to those who still consider me a sell-out.

You may have heard Noam Wasserman's "Rich or King" choice:[39] Company founders are either in it for the money ("Rich") or in it to build a lifestyle and personal identity ("King"). FogCreek and 37signals are built to be "King;" all venture-funded companies are built to be "Rich."

Noam says that successful founders make the "Rich or King" decision up front, and that though it doesn't matter which path you take, you must be consistent in your actions. You can't mix "be king" tactics with "get rich" end goals.

Except I did mix "Rich" and "King," and it worked.

credit[40]

See, it's good to be "King," but what do you do when you're at Trudy's "North Star" Tex-Mex Restaurant tucking into a Chile Relleno (with salsa verde, black beans, and the ground beef filling), and the guy across the table looks you in the eye and **offers you enough money that you never have to work again?**

I was always in it for the money, especially in the form of acquisition. Everyone who came to work at Smart Bear was indoctrinated with this attitude in no uncertain terms; even before hiring someone, I would tell them that we're here to make money, and if someone offers

to buy the company someday, I'm going to sell it, and all of us will make money.

Profit was the rule behind every choice we made. Although the end goal was always acquisition, my attitude was (and still is) that the best way to get yourself acquired is to be profitable.* Profits prove the business is operating well. Profits validate the market. Profits make minimum valuation easy. Profits mean the buyer converts balance-sheet money into bottom-line profit-and-loss money—a trade every large company wants to make.

Most of all, profits mean you don't *need* to sell, which gives you the ability to walk away from a deal. You have little negotiating power in any deal unless you can happily walk away.

On the other hand, I knew I would only be happy building a genuine, great company, where the product solves a real pain, where customers are given white-glove service, where "tech support" is the only sales force, where we leave the world a little better than we found it, and where every employee is smart and gets things done[41] and is trusted with any decision.

And I wanted the ego-inflating trappings of running a company. It's cool at parties to say "I run my own company." I wrote a book (p. 457) that got so popular (in my little corner of the world) that people would bring it up to me to sign. (We gave the books away for free so the joke was that by signing I doubled its value.) When I walked onto a tradeshow[42] floor it was like Norm on Cheers—I knew everyone and they knew me. I got to present at cool venues like the Business of Software Conference.[43]

And I write this blog, shamelessly exploiting Smart Bear's success to convince you that I'm worth reading.

In short, **although the goal was "Rich," I achieved it by behaving like the goal was "King."** I don't know why people find this

* Editor's Note: At this time the blog was written exclusively for bootstrappers with small companies. Obviously there are many examples of companies sold on the basis of growth or "daily active users," but small self-funded companies are most attractive when they are cash machines.

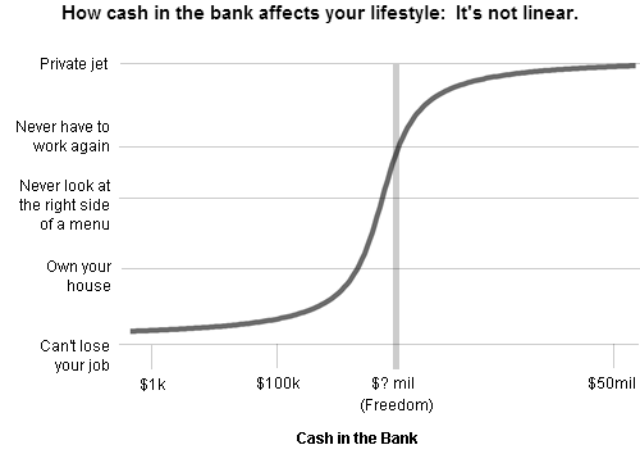


Figure 1

contradictory; after all, acting like "King" means building a long-term, sustainable business, and that's exactly the kind of business that gets acquired.

Still, because "King" was enjoyable and Smart Bear was profitable, I still need to explain why becoming a "sell-out" was the right choice.

The first thing to understand is the non-linear relationship between "cash in personal savings" and "financial freedom" (Figure 1).

There's a line you cross where your savings alone will fund a reasonably lavish lifestyle. At the risk of sounding like George Bush, this is a Freedom Line—freedom from restrictions about what you can do with your life, family, and career.

My observations:

1. A movement from left of the line to right of the line changes your life fundamentally, giving you the freedom to do whatever makes you happy, forever.
2. If you're crossing from left to right, it doesn't matter how far to the right you go. (Sure, $100m is a different lifestyle than $10m, but it's not as critical to lifestyle or happiness as just crossing the line.)

(1) is what was offered to me at Trudy's Tex-Mex. (2) means it almost didn't matter what the offer was, so long as it was big enough.

Some people gave me a hard time about (2). The typical argument was:

> Your company is growing 100% year over year. It's profitable and throwing off cash. Why not wait another year and let revenues double again, which will make the company **six times more valuable** (assuming 3x revenue valuation, a reasonable ballpark for a growing software company).*

Here's the best analogy I've come up with to describe why this is flawed logic. It's called the **Box Game**:

Imagine I have two opaque boxes. Box A contains $10. Box B has a 50% chance of containing $20, and a 50% chance of containing nothing at all. You pick either box and take whatever's inside. Which box do you pick?

Of course statistically there's no difference, so this isn't a question of math or economics or intelligence; it's a measure of your attitude towards risk.

* Editor's Note: This was written in 2009; In 2023, a 5x-10x multiple on this type of company at that growth rate and profit is reasonable.

Most people pick box B. After all, the difference between $10 and $20 is trivial and it's more fun and exciting to pick B.

But what if the numbers were different?

Now box A holds $10,000,000. Box B either holds $20,000,000 or nothing, 50/50 chance. Which do you pick?

You pick box A. Of course! Because it moves you from the left of the Freedom Line to the right. And because a "chance of moving even further" isn't worth giving up the certainty of that life-altering event.

This is my argument in favor of #2 and against "wait and see." This is why I sold.

In my case, the correctness of my choice was made painfully clear by the economic crash in 2008. Had I held out for "another year and far more money"—box B—I would have found an empty box.

I know this for a fact—another company (can't say who, sorry!) was offered a deal at the same time I was. This founder wanted to roll the dice (box B) and delayed the buyer. Two quarters passed and revenue failed to grow; the buyer nixed the deal. Months later with the recession in sight, the founder approached the buyer again, this time willing to accept a low offer. The buyer refused; that ship had sailed.

And I'm not the only one (Figure 2).

Of course there are also those for whom this calculus doesn't apply because they want to be "King" no matter what. I'll bet Jason Fried wouldn't sell 37signals for $100,000,000; neither would Joel Spolsky sell FogCreek. Are Joel and Jason being irrational? Of course not.

But neither was I. Most of us are like me, and Codie, and Andrew (Figure 3).

As of December 2007, I have the freedom to work on any project I want for the rest of my life while simultaneously providing for my family, never again worrying about bills, debt, having a place to sleep, or sending our daughter to any college she wants.

I can stay home with my wife and new baby girl[46] for as long as I want, having all the precious time and experiences and memories that they say money can't buy.

But, in the sense of securing that freedom, it can.

Josh Long
@RealJoshLong

I've always appreciated your take on this and that you were proven right so quickly after making it. I had a friend who passed on a great offer in 2007, only to take a fraction of it in 2012 after a brutal 4 year stretch.

12:15 AM · Jul 24, 2024

credit[44]

Figure 2

credit[45]

Figure 3

And by crossing the line, I did.

Chapter 3:
Extreme brainstorming questions to trigger new, better ideas

10X PRICES · NO CUSTOMERS · NO TECH SUPPORT
MAXIMIZE FUN · RIP-OFF · NO TIME
FLIPPED BUSINESS MODEL · NO WEBSITE
NO MEETINGS · NO CUSTOMER CONTACT
COST IS NO OBJECT · SOCIOPATHIC CEO
MORTAL WOUND · PHILANTHROPIST
ONLY ONE THING THIS YEAR

credit[47]

"Boy, what a day, huh?

How do you generate ideas?

"Brainstorming" is hard—staring at a blank whiteboard, wondering whether someone could make a real-life "dark mode" whiteboard, then realizing that's what a blackboard is, only dustier.

Or the modern version, seventeen pointers flying around a Miro board, zooming to 2000% to read that auto-scaled-down text in each standard-sized virtual sticky-note.

We're blinded by our daily work: No forest, all trees. It's too easy to glom onto an idea that's been knocking around for year, its importance undeservedly increased by the familiarity of repetition. It's hard to think past the last seven customer interviews or support tickets or sales calls. Those are great for generating tactical ideas that fuel our roadmaps, but they are tiny increments that cement our tiny thinking.

The following prompts jostle you out of tiny thinking. Each stretches some dimension of reality to an extreme. So extreme that it is nearly nonsense. But dramatically different perspectives can reveal distinctly new ideas. An idea that would be a 60% solution in an extreme hypothetical case, could be a 2x or even a 10x idea in reality.

Sometimes the extreme is surprisingly appropriate. Unique business models emerge when at least one dimension is so extreme that it defies critics and competitors to even conceive of its possibility.* A fantastic idea fulfilling the right extreme can be a company's entire strategy, unlocking a long-term competitive moat.

It's worth a try.

---

* All of these were considered impossible barriers or business models, until a company did it anyway and won because of it: Zappos' free shoe returns 364 days after purchase, Robinhood's $0 trading fees, Amazon's free delivery with Prime, Netflix's mail-order DVD rental, open-source software companies charging for something that's 100% free and 0% secret intellectual property (now hundreds of billions of dollars in combined public-company value), AirBnB, Uber, and Lyft trusting that strangers will trust strangers in the intimate spaces of cars and homes.

10X PRICES

If you were forced to increase your prices by 10x, what would you have to do to justify it (p. 165)?

What sort of brand looks and feels like something that expensive? What positioning would you take? How would the design of both the website and the product need to change?

What subset of your market would you have to target? Do they have different problems that need solving, or different needs? Would they consider the high price a positive, because it fulfills a need to be seen as someone who is successful, or because they feel "buying from the best" lowers the risk of their decision? What would you need to do for them to be able to display that badge? What sort of relationship would you want to cultivate with each customer? How would your business model (p. 71) change?

What expensive services might you need to supply? Human support? Infrastructure? Is it possible for those costs to be "only" 2x or 3x more than today, so that the net impact is massive profitability?

Often early startups charge too little, and established companies struggle to charge more to existing customers. Thinking about what could justify a massive hike, you might be able to do some of those things and justify at least 2x.

NO CUSTOMERS

If all our customers vanished, and we had to earn our growth and brand from scratch, what would we do?

How would we distinguish ourselves with a unique compelling message, as a fresh competitor in our market? What would we put on

our home page? Would we use a different pricing model to compete better? Would our brand need to change to align with the message?

Would we throw out some features that take a lot of work but not many people actually use (since there's no pesky customers hanging on to the expectation of the old features)? Would we build some new feature that would make us more competitive? Would we change our infrastructure or architecture or UX dramatically because of what we now know, since we're not ladened with existing customers?

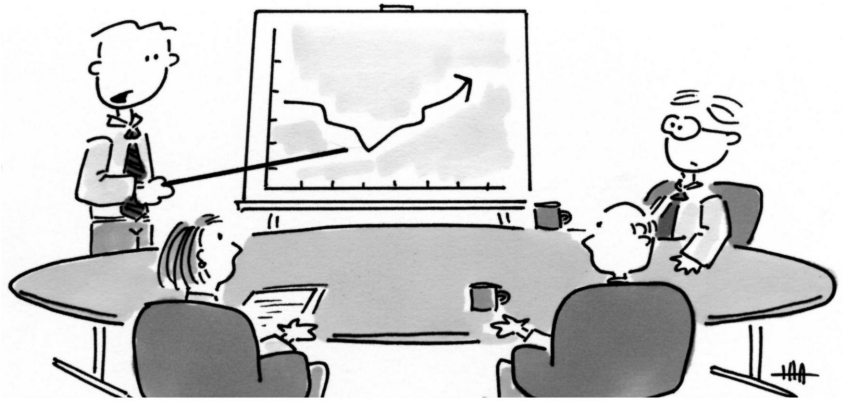
Often we don't make important changes because we've gotten complacent with our marketing, or we never got around to having a truly compelling unique positioning statement, or we don't want to incur the penalty of big changes on our customer base, even if it means we're doing the wrong thing today from a competitive standpoint, and the wrong thing for the future. We don't want to make even 5% of our customers mad, even if it would be better for the other 95%. There will be 10x more customers in the future than there are now, but only if we build for them, today.

NO TECH SUPPORT

If you were never allowed to provide tech support, in any form, what would have to change?

How would on-boarding need to be improved, to the point where customers would self-serve and be happy doing it? This might be the hardest step, because the customer is least-familiar with your product, and least-motivated to power through barriers to their success. And setup stuff often has to be done only once, so you might not learn how to improve those things from existing customers.

Where would the product provide the user with more control, since they can't ask Support to do it for them? What information requires better visibility, since they can't ask Support to look it up for



"So, as you can see, customer satisfaction is up considerably since phasing out the complaint forms."

credit[48]

them? What actions would need to be become more intuitive, because they can't ask Support how to use functionality that they know "is in here somewhere?"

Self-service isn't just better for your costs and scaling, it also makes customers happier and more loyal.[49] *Therefore, dramatically improving self-service is often profitable for both the top- and bottom-line.*

MAXIMIZE FUN

What would be the most *fun* thing to build?

It has to be something that makes our product better, not a random technology or unrelated market. Aside from that constraint, it can be anything—building a feature you just think would be cool for customers to use, developing a technology that would be fun to work on (that happens to deliver a feature customers want), refactoring infrastruc-

ture or architecture using some interesting modern technology that would also improve something like cost-efficiency, scalability, testability, or maintainability.

What if you held a one-week hackathon and actually tried to build one of those things?

When we work on things that are fun, we work better and harder, yet are happy to do it. Therefore, biasing towards fun is also biasing towards productivity and long-term fulfillment, which is practical and logical, not merely indulgent as it first appears. Ultimately, work still needs to advance the strategy while improving something substantial for the customer or the team, but typical prioritization under-appreciates the power of fun.

COMPLETE RIP-OFF

If our biggest competitor copied every feature we have, how would we still win?

Is the answer inside our product, in something other than the utility of the features? What is that, and how could we make more of that, so we're differentiated even in the face of a copy-cat? Is it ease of use, ease of sharing, pleasure of great design? Is it building the next unique feature so quickly that no competitor can catch up (since in this hypothetical they're just copying us rather than being insightful on their own), and if so, what is the killer next feature that would leap-frog us from a customer's perspective?

If the answer is outside of our product, where is it? A higher purpose or brand-promise that our customers buy into, aside from the product? Is that because we are good at communicating how we make the world better, or because our customers are "rooting for David against Goliath," or how our culture is special, or because our brand is distinctive (even if just "it's the best" or "it's the biggest"), or because

our customers trust us, maybe because of our customer service, or because of how we handle situations that go poorly, or because we "give back" in forms like open-source or community-building or significant philanthropy?

All your good ideas will be copied; it's just a question of when. Competing only on features results in bullet-point battles; this is the weakest way to win sales. Creating bigger and more emotional distinction is a powerful way to win, and breaks us of the habit of believing that incremental product updates will dramatically increase differentiation, or growth.

NO TIME

What if we are forced to ship a full, completed MVP (or actually, SLC (p. 101)) new feature, in just two weeks, that would delight and surprise some fraction of our customers.

> *"To achieve great things, two things are needed: a plan, and not quite enough time."*
>
> *–Leonard Bernstein*

Every bit of complexity has to be stripped out—we can add it back later. Scope has to be reduced to a minimum, which reduces how many people can use it on day one, but increases how much we'll learn on day one.

Can we stub out an API, implementing it manually for now? Can we hook up some 3rd-party system, implementing something more efficient later? Can we skip documentation and internal support training and perfect error handling, not forever, but just long enough to see how people like this new functionality?

A 1000-page book can be summarized in a 40-page Cliff's Notes. The key lessons of Cliff's Notes can be summarized in a 1-page blog post. The main theme and one unique perspective of the book can be summarized in one sentence. It can always be smaller, if we trade off detail, or precision, or completeness.

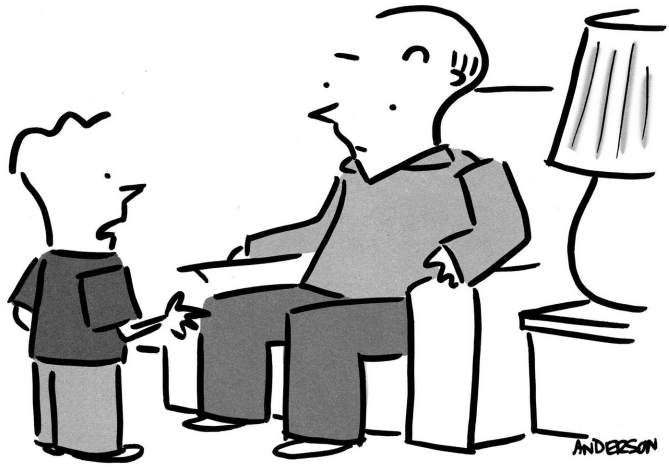
Hackathons prove that we can code really cool things in a short time, when we want to, and if we make certain trade-offs. It's not fully finished, not polished, not ready to scale, but it's something substantial and far more "agile" than teams normally are. If you feel like you've lapsed into a waterfall with two-week report-outs, this might be the shake-up everyone needs.

FLIPPED BUSINESS MODEL

What if you were forced to charge customers in a completely different manner? The product isn't free, but somehow you have to justify value (and costs) differently. If you're usage-based, you're forced to charge a flat monthly rate; if monthly with tiers, you're forced to measure and charge daily with some formula.

How would you have to change how you position the value you deliver? Would you have different kinds of customers due to prices effectively being much higher or lower, and how would that affect brand, messaging, marketing, or sales?

How would costs have to change, so that the vast majority of customers were still profitable? Would software architectures need to



credit 50

"I was thinking – instead of an allowance, how about more of a subscription model."

change? Would marketing or sales structures change? Is tech support suddenly much more or much less possible?

Are there features which you'd need to build (e.g. so customers could control their costs), or features you'd now be able to supply (e.g. because customers would now see the utility in it, or would be happy to pay more because of it)? Would you create features which caused higher prices, but in concert with additional value, so that customers would be happy to grow with you?

Business models force you into certain frames of thinking about how to monetize value that you create, and how costs need to scale with sales and infrastructure. The general idea for any company is to generate far more value than it charges for, so customers are genuinely better off in the exchange, and are happy to be long-term customers. Shaking up the business model shakes up the value/cost equation; sometimes a different business model is actually better for everyone.

NO WEBSITE

If you were not allowed to have a website, how would you still grow your business?

Could you grow by word-of-mouth? Does the product help people do that proactively?

Could you create advocacy, as in channel partners or ambassadors? Could your social media be so good, someone would sign up for the product without needing to know more?

Could your product *be* marketing? Could you cheat, where your "website" is inside the product, where potential customers are caught up learning more and trying things in the moment, so that they become users before they realize what's happening?

Your advocates are already using your product, yet so often we leave "getting more customers" to the marketing department, rather than realizing the product is already a platform for growth. Truly viral products know this already (e.g. you can't use a chat service unless you invite other people to the chat service), but non-viral products can still create growth machines from inside the product. What if "in-product marketing" is more impactful on growth than any useful feature?

NO MEETINGS

What if you made your most introverted teammates' dreams come true: No more synchronous meetings, ever again?

What if you had to on-board new employees without meetings? Or operate the software, deploy the software, find where things are located? What systems would need to exist, and what materials would need to go in there? What format would allow you to update that easily?

What if you had to make decisions without meetings—the process as well as the result of that decision? Deciding what to build, prioritizing big or small things, brainstorming new things? Is it possible that brainstormed ideas could be even better without meetings (because people aren't being influenced by other people, ahead of having their own thoughts) or that decisions could be even stronger without meetings (because people have the time and space to think, to research, to try things out, thus more completely exploring the solution space?).

How would you create social ties and inside jokes and 1:1 as well as group relationships, if you're never in the same space at the same time?

Meetings are still the best way to accomplish certain things: Social bonding, deep discussions where ideas rapidly ricochet off each other, decisions where it's important to "look in everyone's eyes" to get the final agreement, and more. But one of the most common complaints in any company is "too many meetings," or at least, "too many useless/bad/wasteful/inefficient/boring meetings." Besides the usual admonitions about meeting hygiene, a bigger question is whether your systems and processes can not only prevent them, but increase utility beyond that. For example, how much faster could a new team member get up-to-speed if they could read "the why" behind the last 20 major decisions?

NO CUSTOMER CONTACT

If we could never talk to our customers again, how would we figure out what to build?

Could you measure their behavior so well that you could quickly measure whether any change was positive or negative, so at least you could iterate your way to a better product? Are there clear signs of value or happiness that you could use like marketers use conversion with A/B testing?

Could you be even faster at testing new ideas? Could a feature start out as just a button that says "coming soon" or asks "why did you want this" or "what did you expect to happen" or "get emailed when we make this feature," so you could directly measure what people are likely to use, before you invest in building it?

What could you analyze online, in what customers or competitors or observers are saying, that would inform your strategy or even product roadmap? Could you do better than just copying what others are doing? Could you analyze what people are saying about us or competitors online, understanding that the loudest voices aren't a random sample of the population, but still might have something useful to tell us, even when they're not being constructive about it?

Sometimes you interview customers and come away with the wrong feature ideas anyway. Sometimes you ask them whether they'd use something, and they say yes, and they even meant it, but after spending five months building it, they don't actually use it. Stretching your ability to get empirical signal helps you avoid those issues, and might even create breakthrough ideas, or avoid ideas that seem great, but aren't.

COST IS NO OBJECT

What if it didn't matter how unprofitable you were?

Maybe prices stay the same but you can spend 10x or 100x more, even if that's wildly unprofitable. Or maybe the product is free, too. How would you spend the money? How much value could you deliver, if you thought of the product as an exercise in philanthropy?

It could be infrastructure or software, but what about the personal touch? What would a dedicated, personal concierge do? If software could do just 30% of that job, but at 1/100th of the cost, that could be a good feature.

It can be surprisingly difficult to convert money to real customer or business value, even when money is free. If there is a way, it might lead to a new pricing tier, leveraging that idea but adding back a sustainable business model.

SOCIOPATHIC CEO

What if you could change anything, regardless of what anyone thinks or feels?

Terminating an entire product line would shake the organization, and possibly incur layoffs in multiple departments. But what if it's the best thing to do, despite that? What if you allowed yourself to explore what that would be like? What if it turns out you can make that pivot by reorganization rather than layoffs, and now company is executing a much better strategy?

Pivoting the entire company could break the organization, destroy trust in leadership, cause people to leave, and piss off investors. But

"So here's how the restructuring is going to work. I'm going to be Tom. Tom, you're Anne now. Anne will be the copier, and, going forward, the copier is in charge. Questions?"

credit[51]

what if that's exactly the transformation required to fix the strategy? Is there anything that might actually be worth such a move?

You never want to contemplate catastrophic actions like layoffs, whole-company pivots, terminating an entire product line, and for good reason—the consequences are dire, arguably even inhumane. At the same time, that aversion prevents you from thinking certain thoughts. You should at least be able to explore the thoughts. The very best ideas should not be avoided only because they are hard.

MORTAL WOUND

What externality has the potential to kill the entire company?

Is there a company X, which released product Y, at price Z, that would reduce your new sales to a trickle, and would mean half your

customers leave within a year? Can you think of a security breech so significant, most of the customer base leaves within a year? How could the economy change, in which no one would buy your software?

Now, is there something you could do to mitigate that effect? Or something that pivots the company such that it is no longer a threat?

I ran this exercise at WP Engine and we came up with 35 existential threats; it's not wise to worry about all of them, and try to negate all of them. Sometimes, however, you find an idea which is wise to do anyway, that also shores up your defenses. Or, one threat sticks out as being so likely that it warrants acting ahead of time.

PHILANTHROPIST

What if our only goal were to create the most good in the world, personally for our customers?

Ignore how to make money; assume that works itself out. But still talking about our customers (don't argue about which world-wide causes are "most important"). How could we make their lives better, not because our product has greater utility, but because our product is doing something important. How could we help them achieve their ultimate goals (p. 259) rather than just "use features?"

If our customers are small businesses, could we help them grow and thrive? If our customers are writers, could we help them improve their craft? If our customers are performers, could we get them more attention? If our customers are enormous companies, could we help them use their greater scale and wealth to create more good in the world? If our customers come from all walks of life, could we increase the access and equity in their reach and power?

There's already strong evidence that having a mission bigger than yourself, more than metrics, creates more—and more loyal—customers and ful-

filled employees (p. 399) both. Recent surveys of Gen Z show that they care about whether a company does good in the world just as much as they care about compensation. So, even aside from the ethical reasons, there are "self-interested capitalist" reasons to create more good, not just more MRR.

ONLY ONE THING THIS YEAR

What if you could only ship one thing this year?

What single initiative would make the most difference? What's so impactful that it would actually be OK if we shipped nothing else? You get the entire year, so there's time for a release, and fixes, and enhancements, but it has to be only one topic.

What would generate the most revenue? What would be so differentiated in the market, and so desirable, that customers still buy and stay even with your lack of any other substantial product changes? What would be so exciting that customers would stay even if other things are missing or have on-going bugs?

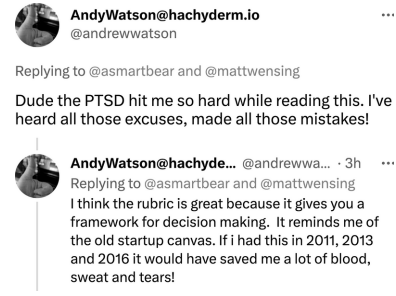
When you look back over years of a business, often the whole trajectory comes down to 1-2 big decisions per year. A critical product launch, a key decision to enter a market for expansion or exit one market to focus on a healthier one, a key hire, a competitive insight. It's difficult to know in the moment which will be seminal, but thinking this way forces you to think of only the absolutely most impactful ideas, which are probably the ones you should be focused on regardless.

> *"The dangerous man is the one who has only one idea, because then he'll fight and die for it. The way real science goes is that you come up with lots of ideas, and most of them will be wrong."*
>
> *–Francis Crick*

Chapter 4:
Excuse me, is there a problem?

THE PATH · PLAUSIBLE · SELF-AWARE · LUCRATIVE
LIQUID · EAGER · ENDURING
EVALUATING VIABILITY · FURTHER READING

How many companies fail:



credit 52

1. Founder gets a flash of insight: The world has a Problem.
2. Founder talks to three potential customers who are experiencing The Problem, or who are expert in the domain of The Problem. They agree The Problem exists. *(And they're correct!)*
3. Founder builds a product that solves The Problem. *(And it really does!)*
4. Founder fails to make enough sales, and the company shuts down in 6-24 months, when the founder runs out of patience or money.

5. Founder laments into the void (i.e. posts on Twitter): Why were sales so hard when the product clearly solved a real problem?

These companies fail because solving a problem is—perhaps surprisingly—not nearly enough to build a successful company.

The following model explains an extremely common reason why this happens, and what you can do about it.

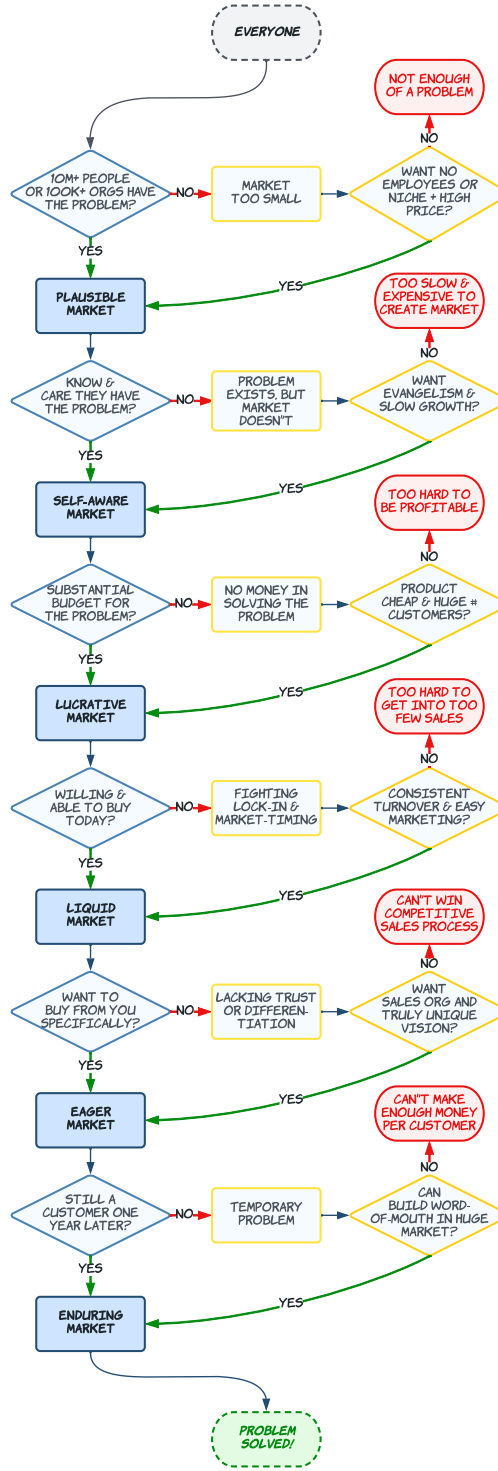
In the discussion, you'll figure out where your challenges are, and whether you can design a strategy to side-step the issue, or whether your business simply isn't viable.

> *"Greatness needs luck, but it's never by accident."*
>
> *—Unknown*

THE PATH FROM "THE PROBLEM" TO "VIABLE BUSINESS MODEL"

The main challenge facing a new startup is that so many different things have to go right for it to succeed. A subset of those things is the path "Problem" to "Viable Business Model."

Let's dive in.


EVERYONE
10M+ PEOPLE OR 100K+ ORGS HAVE THE PROBLEM?
NO
MARKET TOO SMALL
WANT NO EMPLOYEES OR NICHE + HIGH PRICE?
NO
NOT ENOUGH OF A PROBLEM
YES
YES
PLAUSIBLE MARKET
KNOW & CARE THEY HAVE THE PROBLEM?
NO
PROBLEM EXISTS, BUT MARKET DOESN'T
WANT EVANGELISM & SLOW GROWTH?
NO
TOO SLOW & EXPENSIVE TO CREATE MARKET
YES
YES
SELF-AWARE MARKET
SUBSTANTIAL BUDGET FOR THE PROBLEM?
NO
NO MONEY IN SOLVING THE PROBLEM
PRODUCT CHEAP & HUGE # CUSTOMERS?
NO
TOO HARD TO BE PROFITABLE
YES
YES
LUCRATIVE MARKET
WILLING & ABLE TO BUY TODAY?
NO
FIGHTING LOCK-IN & MARKET-TIMING
CONSISTENT TURNOVER & EASY MARKETING?
NO
TOO HARD TO GET INTO TOO FEW SALES
YES
YES
LIQUID MARKET
WANT TO BUY FROM YOU SPECIFICALLY?
NO
LACKING TRUST OR DIFFERENTIATION
WANT SALES ORG AND TRULY UNIQUE VISION?
NO
CAN'T WIN COMPETITIVE SALES PROCESS
YES
YES
EAGER MARKET
STILL A CUSTOMER ONE YEAR LATER?
NO
TEMPORARY PROBLEM
CAN BUILD WORD-OF-MOUTH IN HUGE MARKET?
NO
CAN'T MAKE ENOUGH MONEY PER CUSTOMER
YES
YES
ENDURING MARKET
PROBLEM SOLVED!

PLAUSIBLE: DO 10M PEOPLE OR 100K COMPANIES HAVE THE PROBLEM?

These numbers sound larger than necessary, but here's why it *is* necessary even for an indie startup, using Fermi Estimation (p. 171):

1% conversion: Impression → Visitor
An AdWords campaign with multiple keywords, ads, and bids would be very successful at a 2% click-through rate. Display ads are more like 0.3%-0.5%. *(source: HubSpot[53])*

Top SEO position can be 3%-5%, but that's almost impossible to achieve for even a mildly competitive keyword. *(source: HubSpot[54])*

1% conversion: Visitor → Paid
A typical, successful product website converts 1% of its traffic to paying customers. I don't have firm data, though I did an informal poll on this question years ago. Some[55] data[56] show 2-5% conversion rate even for just a sign-up form or free trial, of which a fraction will become paying customers.

Therefore: 10,000 Impressions → 1 paying customer
10,000 × (1% click-through) × (1% convert-to-pay) = 1.

10,000,000 Impressions → 1,000 paying customers
Not every impression will be a unique person, but you still need closer to 10M potential eyeballs than to 1M, because while some people will see your material more than once, most of the market will never see your material.

1,000 paying customers is the minimum* needed for a sustainable, small company. It will take about two years,** 10M marketing impressions, and luck (p. 1035).

---

* This is a rough rule of thumb companies charging $30-$100/mo; if less, you'll need even more customers to become sustainable; if more, you need fewer cus-

If you're selling directly to consumers, there needs to be 10M who have the problem, otherwise it's too small even for an indie company who wants to stay small forever.

If you're selling to businesses, the total number of potential customers is an order of magnitude smaller, but they will pay orders-of-magnitude more to solve problems, and conversion rates are higher, thus 100k is sufficient.

A lot of great ideas attack problems that just aren't actual problems, at least not for more than a small handful of people, and therefore fail to yield a successful company.

Can you be successful anyway? Yes, there's an exception to every rule. For example, a high-price-tag product in a small niche can be a fine company. Or, perhaps you're happy staying frugal, never hiring an employee, and making $100,000/year post-expenses post-tax, replacing a salary but on your own terms.*** That's wonderful. You can be the exception, but with conditions.

---

tomers, but they are more difficult to find and convert than the numbers above suggest. Pricing determines your business model (p. 515).

** WP Engine, the company I started over a decade ago, was a hyper-growth company and then a Unicorn, reaching $100M in ARR in a similar time frame to other hyper-growth companies (as in the chart mid-way through this article (p. 193)) now with over 200,000 customers, yet it took two years to get the first 1,000 customers. Competitors who started after us also took that long, and there[57] are[58] many[59] many[60] many[61] many[62] many[63] many[64] many[65] many[66] many[67] many[68] other[69] examples.[70] Of course it can also take forever,[71] 4 years,[72] 6 months,[73] or 52 hours,[74] because this is a rule of thumb, not a law of physics.

*** This is, in fact, what 80% of small businesses[75] do. It is a vibrant and valuable driver of fulfillment (p. 399) and the economy; ignore those who claim this is somehow less important than "swinging for the fences." Rather, there are two kinds of companies: Those which endeavor to replace a salary (and then some), and those who are trying to become huge, and they are simply different paths.

SELF-AWARE: DO THEY KNOW & CARE THEY HAVE THE PROBLEM?

It seems like the answer should be "obviously yes," but often the answer is "shockingly no."

If a person does not already believe they have a problem, they will not be surfing the Internet looking for a solution, and even if they happen upon your website somehow, you cannot get them to spend money to solve a problem they don't think they have.

Sam Altman is the co-founder and CEO of OpenAI, and before that ran the Y-Combinator accelerator, and is therefore one of the world's most experienced experts on startups. He previously co-founded Loopt—a location-based, mobile social-network app. Oh look, those are *all* the keywords of 2005, when it was founded. It raised $30M and failed. When asked what happened, he said:

credit[76]

> The market wasn't there. You can't force a market. You can have an idea, and as a startup part of your job is to be ahead of it, and sometimes you're right about that, and sometimes you're not.
>
> Sometimes you make Loopt, and sometimes you make OpenAI. You just keep trying.
> —Sam Altman, interviewed by Kara Swisher[77]

I've given the example (p. 239) of website security, which I know* is a real problem plaguing millions of websites whose owners think "those mean hackers won't attack lil' ol' *me*; I'm nobody!" False. Hackers indeed don't care about lil' ol' *you*, but they do want to gain control of your lil' ol' *server*, so they can do their nefarious things, like spamming, advertisement-click-fraud,[78] remote-controlling[79] your visitors' browsers, or just bouncing off to yet another server as a way of covering their tracks. Everyone has the problem, but millions of people don't *think* they have the problem, so they're not searching for website security software, and certainly not buying it.

"Security" is a case of ignorance, but the other version of this challenge is when the customer knows they have the problem, but genuinely does not care. This could be because this problem is the ninth-most-important priority on their list, and they can only give attention to their top three… and this item will never bubble up to the top three. An example I see a lot at WP Engine is accessibility.** Given lip-service by many marketing departments and product managers, it rarely makes the priority list for the public website or the product.

---

* A million websites run on WP Engine's platform, serving tens of billions of requests daily, as 9% of the global online population visits a WP Engine property every day. We block hundreds of millions of nefarious requests daily, and internally run SOC Type II and ISO/IEC 27001:2013 certified security processes. So we know a lot about what hackers do to websites large and small.

** "Accessibility" means working well for people with various challenges; in the visual sphere, consider cases like red/green color-blindness, needing high-contrast colors, needing larger text, or complete blindness, needing "screen reader" browsers to navigate menus, forms, and content.

Some companies choose to care, or are mandated by governments or contracts. This author chooses to care.* But the fact is, though everyone agrees they have the "problem" of a non-accessible website or product, most don't have the will to act.

Sometimes "willing and able" is a matter of market-timing. A famous example is Instacart: Successful after 80% of Americans carried a smart phone, unlike WebVan[80] which was exactly the same idea, solving the same problem in much the same way, but the market wasn't ready for it.

A lot of great ideas, attacking real problems, fail to become successful companies, because the target market doesn't know they're even in the market. Because they're not.

Can you be successful anyway? Yes, there's an exception to every rule. Some founders are not only the first-and-best sales-person, but also natural evangelists. More, they're on a mission to educate the world about their passion. They don't see a lack of interest as a barrier, but as an opportunity to change minds. That is a difficult, expensive, and slow path,** but it is a path, and one that could result in zealous, loyal customers and a fulfilling existence. But you really have to want that path, doing that work with those consequences, if you're going to enter a market that you also have to create.

---

* This site uses semantic tags for content and navigation, the entire stylesheet of both fonts and layout supports arbitrary changes in font size, automatically respecting browser-specific settings, supplies keyboard shortcuts for menu actions, has alt-text for all images, uses `aria-title` and related attributes, uses high-contrast foreground/background colors, also supports high-contrast mode, and works well in screen-readers (including those built for everyone, like Safari Reader and Readwise Reader and Pocket). That said, let me know if there's more I could be doing!

** "Difficult" because changing someone's mind about anything is almost impossible, especially when they're not seeking to have their mind changed. "Expensive" because of the marketing and attention you have to bring to the cause, on top of the usual work of making a sale, with certainly-worse-than-average conversion rates. "Slow" because you're having to create demand and then fulfill it, rather than meet demand that already exists.

LUCRATIVE: DO THEY HAVE SUBSTANTIAL BUDGET TO SOLVE THIS PROBLEM?

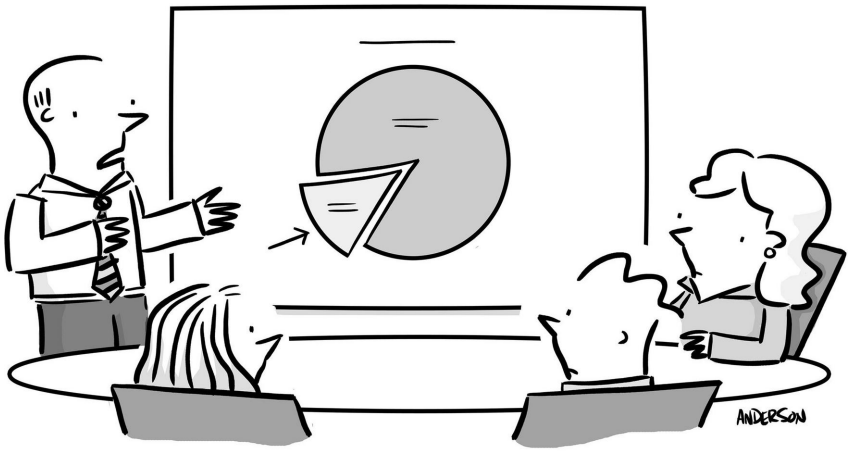
- *"You have a great product! I'd have to get [someone else's] approval for this though, and they don't understand all this."*
- *"This is nice, I would use it, but they're carefully watching all expenditures and the truth is I can manage without it."*
- *"I'd love to, but our budget is closed for the year and I can't start a new project."*
- *"This is pretty cool, but our internal team who manages [the problem] says they don't need help. They might be just trying to save their jobs, but it is what it is."*

We've all heard these objections. Some are normal; you can't win every sale. But sometimes the target customer agrees they have the problem, yet doesn't have the money to solve it.

At Capital Factory,[82]* there's a constant stream of kids coming out of college with a startup that "sells _____ to college students." It's easy to find "problems"—restaurants and bars want to advertise to students, students don't want to spend much on food, students need books and supplies, and so on. The founders explain they "had the problem themselves, so they *really* understand it."

That's probably true. The deal-breaker is that college kids have no money, and don't spend what little money they have on SaaS products. And businesses that cater to college kids have to charge low prices (because college kids have no money), and therefore don't have budgets

* Capital Factory is by far Austin's largest and most prolific "Center of gravity for entrepreneurs in Texas," now a tiered system from co-working to mentoring to multi-million-dollar investments with hundreds of companies in orbit. The University of Texas, also in Austin, has a high-quality Computer Science department that is also one of the largest in the country.



"Every year there are naughty children who don't get any toys. That's a vastly underserved niche I think we can capitalize on."

credit[81]

for oddball new ideas. In 16 years, none of these startups worked, even though arguably most of the "problems" existed.

You might think a large company will definitely allocate budget for a known problem, but here again the answer is often in the negative. Budgets are applied to the top few most-important problems of the year; if this is a problem, but not a top one, it won't get attention. Large companies have to allocate more than money—they have compliance teams who have to approve, they compare multiple vendors, they run pilot programs, and all of this won't be set in motion unless it's a top problem.

Large companies have internal teams that are already tasked with the problem, which might means there's no additional budget for outside solutions. Those teams often fight against outside tools that are seen as making their jobs obsolete, or at least converts them into vendor-managers instead of innovators. You want to target companies who outsource this particular problem to outside vendors.

Once you get over the hurdle of there being a budget at all, is the budget large enough? I'm always shocked how little people will pay for productivity applications like to-do lists and note-taking. These are applications you might use dozens of times per day, as much as email. The slightest increase in efficiency or even simple delight will have a massive impact on the customer's life, every day. And yet people complain that the Pro version of Bear App[83] * is a whopping $15/year, or that Remember the Milk[84] is $40/year, or how they've been paying €29/year for Evernote for eleven years, but a change to €43 is so devastating that they will completely change to another application that has 1/10th the functionality and no tech support (Figure 1).

In general, consumers don't like paying for stuff, hence the multi-trillion-dollar success of having people "pay" with attention (advertisement) and data (privacy). This is why I think** self-funded companies in particular should target businesses as customers; unlike consumers, they will spend money to solve problems and to make more money (p. 165).

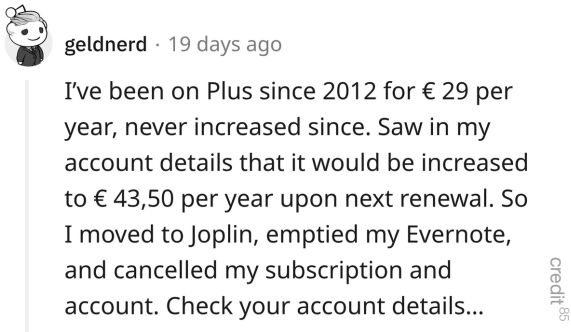


Figure 1

* My note-taking application of choice, and not just because it's called "Bear!"

** This is clearly a personal bias. I can't wrap my head around the mentality described above, and that's why I don't build in and generally don't invest in B2C—I'm fully aware that I don't understand the customer!

A lot of great ideas, attacking real problems that customers acknowledge and seek solutions for, are in areas where budgets don't exist, or not often, or are so small that it requires an enormous number of customers to make money (often also in a crowded competitive space), and therefore the company fails.

Can you be successful anyway? Yes, there's an exception to every rule. If there are a huge number of potential customers, and if your cost-basis is extremely low,* you can create a strategy targeting a large market at a low price with a simple product.** This works even better if existing products are poor (so you can stand out and make a splash) and expensive (so your low price is itself a differentiator). It's still risky and difficult, but you could accept that and make decisions consistent with that challenge. But you have to really want that path, and increase the "10M" number, since you'll need a lot more customers to make ends meet.

LIQUID: ARE THEY WILLING AND ABLE TO BUY RIGHT NOW?

- *"Oh yeah, we spend $100,000/year on this. But our contract isn't up for renewal for another 15 months."*

---

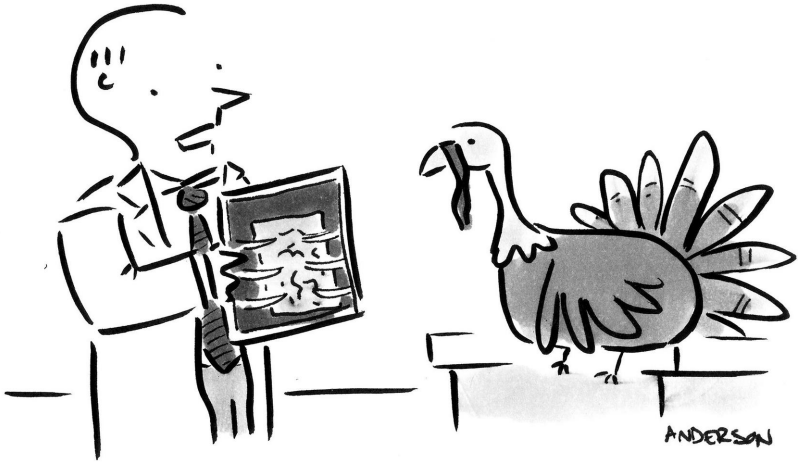
* e.g. no direct customer service, no substantial infrastructure costs, a route to market that costs almost nothing to acquire a customer, the ability to build and maintain the application with very little design or engineering or product outside of the founders, switching costs are low so you don't have to do a lot of work or spend a lot of money to get a customer off another product and on-boarded onto your product

** When a product will be widely used with little-to-no customer service, it must be simple or it won't scale.

- *"We just implemented a version of this in Workday. I'd rather use your product actually, but it's just part of Workday and the HR team likes that everything is integrated."*
- *"This is way better than our current system, but we've invested a lot integrating with seven other systems, plus a few custom things some engineering teams did, so we can't really consider switching away at this point."*
- *"That looks like great software for weddings. I'll let you know when I get married again! Haha!"*

When the customer is already paying to solve the problem, or actively comparing options to solve the problem, you still run into the barrier of whether they have the organizational will to buy from you.

This can be for legal reasons, like being locked into a long-term contract or government fiat. This can be for convenience, as in the Workday example which at WP Engine caused us to cancel several other SaaS products because "now it's all in one system, which we're



"We found the problem – all of your internal organs are in this little bag."

credit[86]

paying for anyway, so this is simpler and safer to manage." This can be because of other forms of lock-in, like difficulty in extracting and moving existing data, having to retrain thousands of employees, or having to re-implement a variety of cross-systems integrations that people rely on for their workflows, data, reports, and governance.

Notice that **all these forces have nothing to do with your product or its price**. They are so powerful, they overwhelm a product that is solving the problem the best, at the best price. That's why they cannot be ignored, and why founders are surprised when their genuinely-great product in the definitely-extant market where customer are definitely paying for solutions, is still too difficult to sell.

On the bright side, this is a prompt for your strategy. How will your strategy create these sorts of "lock-ins" that will prevent your *competitors* from kicking *you* out of *your* customers?

A final way that customers might not be able to buy right now, is when the product is needed at a specific moment in time, but not before or after that moment. Websites for events and occasions are an example, as are tools that solve temporary problems like sophisticated code profilers or load-testing tools. Because Smart Bear* was in the developer tools market, I know a number of founders of products in the latter two spaces; all of them struggled to maintain even small companies, exactly because people didn't proactively need the product (i.e. "Don't have the problem"), but then suddenly did (i.e. "Willing to buy, but only *right now*"), and then didn't any more (i.e. "No longer willing to buy").

Or, that moment in time might be in an ill-defined future, when the problem is of higher priority. If you're not one of the customers' top three priorities, they can't devote the time or budget to it, even if they agree the problem exists and that your software will solve it for a reasonable price. This article explains this effect in detail (p. 479).

Can you be successful anyway? Yes, there's an exception to every rule. If there's a legal contract, you can offer to pay the penalty

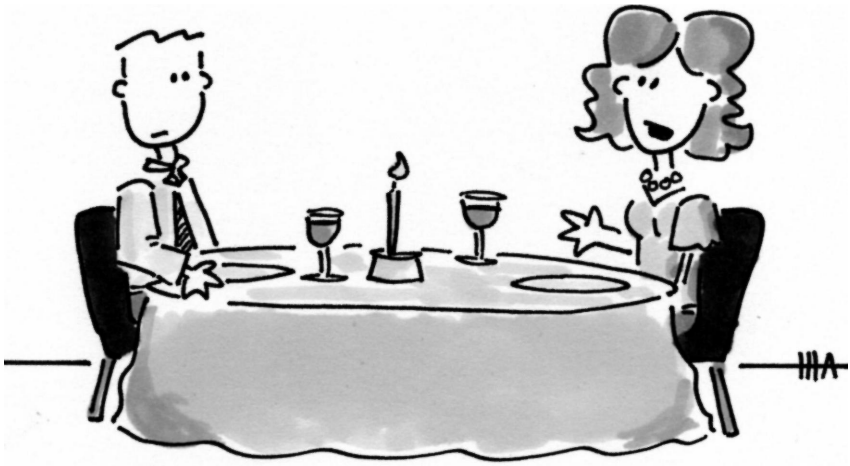
* My company before WP Engine and the namesake of this website.

for them breaking their contract (so long as that isn't so expensive that you can never be profitable). If it's hard to migrate, you can offer to do the migration as a service, perhaps even for free (this is common in WP Engine's market). If you're fighting the all-in-one enterprise systems like Workday, you can focus instead on a target market that cannot afford Workday and sees Workday as overly complex and cold, so that Workday isn't even a competitor. If timing is important, you might offer a way to buy that is cheap or even free while not using it, so you're "right there as soon as you need us again." But you have to really have answers for these challenges, and be ready to walk that path.

EAGER: DO THEY WANT TO BUY FROM YOU, SPECIFICALLY?

- *"It seems like it would work for us, but it looks like you've only been in business for a year?"*
- *"It definitely worked fine during the trial, but we're expecting to grow 100x and we're not confident that you'll be able to handle it."*
- *"For the features we need, [competitor] looks the same to us, and they're cheaper."*
- *"I like how you do X, but [competitor] does Y and Z, which we really like, so we're going with them."*
- *"Our policy requires that all vendors are SOC 2-compliant and provide a security audit trail API, so you did not meet our basic requirements."*

Even in a real market, with customers spending real money, in a purchasing process right now, you haven't yet won. There are other options, ranging from direct competitors to indirect alternatives. They will buy, but will they buy from *you*?



credit[87]

"It's not that I don't like you, Ted, you just don't fit my target demographic."

The first hurdle is **trust**: Do they trust not only that the product works, but that your company will be around for many years to come, that you will maintain a high pace of development, that you won't have security issues, that your customer service will truly help them when inevitable problems arise, and that you can scale as the customer's needs scale?

The second hurdle is **differentiation**. This doesn't just mean "you have something unique (p. 891)." You might have a feature that no one else has, but if only 10% of the market cares about that feature, that's not enough. Worse, your competitor will have some feature *you* don't have, and what if 30% of the market cares about *that* one? How to do this? See this companion article on leverage (p. 543).

A special difficulty comes when the product over-serves a large segment of the market. This means that, for example, you have ten features, but most of the market really cares about three of them. You might have all sorts of differentiation in the latter seven, but that won't sway most people. Picking something simpler or cheaper is the

rational choice, even though your "feature comparison matrix" shows that you're a much more complete solution.

A sales team can combat both of these challenges. The job isn't just to schedule calls, cajole potential customers into action, and press for the close; the job also to build trust in your organization and talk around competitive points, positioning so that the customer wants to buy from the mixed bag of plusses-and-minuses that every product contains.

The best way to overcome these challenges are the "Love" and "Utility" types of willingness-to-pay (p. 275).

Can you be successful anyway? Yes, there's an exception to every rule. Trust can be side-stepped by building a type of product that doesn't require much trust.* Or trust can be built by mitigation, for example open-source means the customer can shift to another vendor or take over the code base in a worst-case scenario. Differentiation is harder. Sometimes you're competing on price, which isn't ideal for the bottom-line or for the quality of customer, but can work very well. In a large market, differentiation can come from specializing in a niche. In a small market, you might not have many viable competitors, lessening the importance of differentiation. A company mission that is "bigger than all of us" can also be a distinguishing reason to buy,** although typically more for consumer products where that's an allowable purchase-criterium. Still, it difficult to survive when you're no different from more mature, feature-rich, stable, innovative alternatives.

---

* "Security" isn't a concern if the data in your app isn't private, such as a social media management tool. "Uptime" isn't a concern if the product is run locally or the service isn't time-critical. "Company maturity" isn't a concern if it's a tool for individual use; in fact it can be an advantage for a new startup to sell to freelancers or other people who want to support other startups.

** People buy windbreakers from Patagonia that are undifferentiated from other outdoor apparel vendors, because Patagonia is well-known for spending hundreds of millions of dollars on conservation and sustainability, and for treating their employees well.

ENDURING: WILL THEY STILL BE PAYING (OR PAYING-IT-FORWARD) A YEAR FROM NOW?

I cannot count the number of indie developers who grow to $15k MRR and start slowing down because their cancellation rate is 7%. Many don't think this is a problem; many of the rest believe that 5% would be a success. It's not.

Financials and SaaS metrics (p. 645) aren't even the primary reason why this is fatal. The problem is that the customers don't want the product. 5%/mo cancellation means only half the customers will still be customers a year from now, which means you're not building a sustainable business.

The reasons can vary. Perhaps they needed the product temporarily; they might have loved it, but "the problem" disappears. More often, the product didn't work well enough—insufficient features, too many bugs, didn't integrate with some other system, too expensive for the end result, turns out the problem is not important enough.

The financial reality is illuminating too. The challenge is that top-line growth is linear for many companies, and quadratic (*not* exponential) (p. 115) even with hyper-growth companies. But cancellation *is* exponential—that's why it's measured as a percent of the current customer base. Exponentials grow faster than lines or quadratics, therefore cancellation "catches up" faster than you can add new customers. Growth fades, and finally ceases, as one customer cancels for every new customer who signs up.

With $15k MRR, adding $2k/mo of new customers—a healthy 15% *per month* growth rate—a 7% cancellation rate means already half of that growth is negated by customers leaving. The company barely got started and already its growth is being decimated. At that rate, only one year later, having grown to about $27k MRR, the company has stopped growing completely (Figure 2), despite spending time and

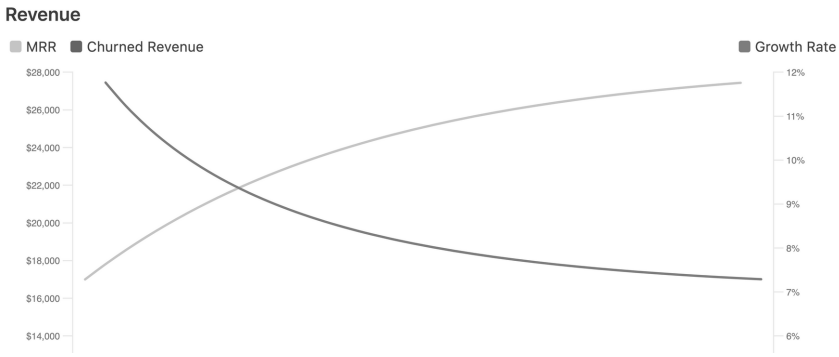


Figure 2: Charting adding $2k/mo new MRR with 7%/mo using the SaaS Plateau Forecaster[88] from Summit.[89]

money on marketing and sales.* Don't forget—those new customers cost money to attract, sell, and on-board with tech support, but all the value of that expense is negated by an equal number of customers walking out the door.

You certainly aren't required to have a goal of "grow forever,"** but capping growth because customers don't really value your product, is not a healthy business no matter what your end goal is.

This example was for a recurring-revenue business, but the same principle is true for one-time revenue businesses. **One-time revenue businesses still require repeat revenue**, in two ways:

1. Customers buying again.
2. Happy customers telling other people to become customers.

---

* Growth stops because the $2k of new customers arriving in a month are negated by $27k × 7% ≈ $2k customers cancelling in that same month.

** But especially if you do have the goal of becoming a scale-up Unicorn, it is impossible to do that without low churn. As just one example, Gainsight CEO Nick Mehta recently pointed out why "This stall will happen to all companies eventually," explaining[90] that launching a second product was their solution.

Both of these require that customers are happy with the product. This reinforces the point that the most important problem is that the customer isn't satisfied, regardless of business model.

Can you be successful anyway? Yes, there's an exception to every rule. Shopify is a fantastically successful business with a 8% cancellation rate—only 34% of new customers stick around for one year.[91] They're successful anyway because (a) the ones that do stay tend to grow forever and super-linearly, (b) the market is enormous and still growing fast, so they won't run out of new customers anytime soon, and (c) customers cancel because their own business didn't succeed, not because Shopify's product is problematic. If the problem really is the product, I don't believe there is an exception. The product has to be work well for some segment of the market. The rest can be mitigated if other factors of the market, strategy, and business model can overcome this massive weakness with even-more-massive advantages.

EVALUATING STARTUP VIABILITY

This path of "problem" to "business model" is not the only factor that determines success. There are still questions like whether you can reach customers, get their attention in the noisy Internet and competitive space, can you do that cost-effectively, are company costs too high, do you have the skills, can you hire for skills, do you have enough time and money to do it, and so on.

Still, we can evaluate the viability of this path with the following model. We'll use Fermi Estimation (p. 171) to avoid the analysis-paralysis of deep research and arguing over details.

Some people have already started using this to think through their businesses, like Sam Bhagwat,[92] co-founder of Gatsby,[93] think-

ing through[94] his next startup. There's also some online calculators[95] if you like that sort of thing.

Answer with regard to a *specific* target market, which means a specific type of buyer solving a specific problem with a product that has made specific trade-offs, at a specific price.

Criteria	Score
Plausible Number of potential customers (consumers or businesses) who actually have the problem	*Power-of-ten only* `1k`, `10k`, `100k`, `1M`, `10M`, `100M`, `1B`
Self-Aware Willing to solve the problem	`0.01`: Few agree or care `0.1` : Thought-leaders care and evangelize `0.5` : It's an industry standard-practice `1.0` : Almost impossible to find someone who doesn't care
Lucrative Annual *allocated* budget	*Power-of-ten only, of net-revenue** `$1`, `$10`, `$100`, `$1k`, `$10k`, `$100k`, `$1M`
Liquid Frequency of purchase decision	`0.01`: Every few years `0.1` : An annual decision `1.0` : Always in the market, easy to switch

* "Net-revenue" means your revenue after subtracting pass-through costs. For example, an eCommerce platform might process a $100 purchase on behalf of its customer, keeping $10 for itself as payment; that is net-revenue of $10. Pass-through costs do not include cost-of-goods-sold, i.e. you should not subtract out the marketing and sales costs to acquire the customer, nor customer support, nor infrastructure costs for SaaS products. Those are important for profitability, but in this exercise we are focused only on top-line revenue, not on the efficiency of your operations.

Criteria	Score
Eager *(identity)* Attitude towards your company	`0` : They cannot buy from you `0.1` : Structural challenges `0.5` : Indifferent; no red flags `1.0` : Mission-level emotional desire to select you
Eager *(comparative)* Competitive differentiation	`0.1` : No material differentiation `0.5` : Some features are so good, some people will buy just for that `1.0` : One-of-a-kind solution that has no viable alternative
Enduring Will they still be here a year from now?	`0.01`: One-off purchase without loyalty `0.1` : One-off purchase, but happy customers will buy again and tell their friends `0.5` : Recurring-revenue from a recurring-problem `1.0` : Strong lock-in from fiat, integrations, or being the system-of-record for a business-critical system
Normalize Normalize the score so that `1` is the threshold for an indie startup, `2` or more for a scale-up.*	*Divide by* `625,000`

Now you multiply. Why multiply? Because this is a series of "ands"—there needs to be customers with the problem *and* they have budget *and* they are buying today *and* so on. The effects compound.

* Justification: Using the figures earlier in the article, you could be successful with 10M consumers at $10/mo, or 100k businesses at $1000/mo (e.g. dentist practice-management), so consider the threshold of those numbers combined to be 100M. Taking the middle value of all other questions—neither a deal-breaker nor a strong advantage—you end up with `100M × 0.5 × 0.1 × 0.5 × 0.5 × 0.5 = 625k`. Arguably you should have some strong advantages, but also some of these values will be on the low side, so this is a reasonable Fermi-style acceptance threshold.

This is dangerously close to a silly quiz, so we have to be careful to use the final score as guidance rather than precise analysis. Still, different choices of target market, target customer, and product trade-offs can result in dramatically different results.

As usual, **having to think through the answers and trade-offs is most of the value of the exercise**, even more than the final score.

A few examples

Still, let's try it, using WP Engine* as the example. Note how our research is simplistic, but because we only have to be accurate to a power of ten, the answers are easy anyway:**

Criteria	Score	Justification
Plausible Number of businesses who have the problem	`100M`	There are 334M[96] businesses in the world, 71%[97] of which have a website. 43%[98] of websites are WordPress.
Self-Aware Willing to solve the problem	`0.1`	While everyone using WordPress definitionally hosts it somewhere, and industry practitioners often use a specialist vendor, most target customers don't care enough to do more than the bare minimum.
Lucrative Annual allocated budget	`$100`	Difficult to say over such a large market; one could argue that `$10` is more accurate because most businesses are small and most don't buy expensive American things.

* WP Engine is the largest platform for WordPress-based websites among the top ten million websites in the world. Therefore, we'll take the market as "businesses using WordPress."

** For example, our figures for the first row are just page-one Google search results, but even if the figures are off by 50%, certainly the true answer must be far larger than 10M and far smaller than 1000M, hence 100M is the easy choice.

Criteria	Score	Justification
Liquid Frequency of purchase decision	`0.01`	People rarely change their website platform.
Eager *(identity)* Attitude towards your company	`0.5`	Could argue that it is `1.0` today because of our leadership position, but ten years ago we were one option among several, all of which were viable for many customers.
Eager *(comparative)* Competitive differentiation	`0.5`	We have many capabilities and features that are either unique or we are the best, but competitors have other advantages, whether in features or price or geography.
Enduring Will they still be here a year from now?	`1.0`	Today we easily justify this with our world-class retention metrics across 15 years of customer data, however even early on we saw high retention, and comparables also have high retention.

Score = `2,500,000 / 625,000` = **4**, so it qualifies as a scale-up, and indeed that's what happened.

Let's try it again with an indie startup: ConvertKit,[99] a email marketing product competing with giants like Constant Contact and newcomers like Substack, designed to help you grow and then monetize your subscribers. They target creators who want to monetize their newsletters, not "anyone and everyone," which reduces the target market but increases willingness-to-pay.

Criteria	Score	Justification
Plausible Number of people who have the problem	`10M`	There are tens of millions of newsletters using competing products.

Criteria	Score	Justification
Self-Aware Willing to solve the problem	`1.0`	This is well-known as a best-practice, with multiple at-scale companies serving a mature market.
Lucrative Annual allocated budget	`$100`	Customers with complex workflows and many thousands of subscribers will pay more, but most customers aren't in that category, and might pay $9/mo.*
Liquid Frequency of purchase decision	`0.01`	Customers buy newsletter software and then want to just use it, not be switching
Eager *(identity)* Attitude towards your company	`0.5`	The founder was well-known among other indie founders, however in the broader market there's no reason not to trust them, but no particular mission-driven reason to pick them either.
Eager *(comparative)* Competitive differentiation	`0.5`	Most features are similar, but there are interesting things at the margins for routing readers through flows and monetizing content, whereas more generic newsletter products don't have as many tools for direct monetization.
Enduring Will they still be here a year from now?	`0.5`	For the customers who avail themselves of complex automations, this would be a `1.0`; most people will probably use it as a normal newsletter, with a simple follow-up flow that competitors also provide.

Score = `1,250,000 / 625,000` = **2**. This is a good business model, possibly even a scale-up, and indeed ConvertKit grew quickly

* Alternately, you could focus on the customers who pay more; you might then reduce the "number of people who have the problem" to `1M`, and the annual budget to `$1000`. While results in same score, it's a different product, serving a different market, so this is an important decision.

as a bootstrapped company, and while not a Unicorn, is an order of magnitude larger than most small businesses ever become.

Finally, let's take the case of selling security software to consumers. This is a tough market; there are success stories (e.g. Norton, 1Password, Cloudflare), but it's hard to find small indie companies who are successful in this area (whereas it's easy to find successful indie WordPress hosting companies):

Criteria	Score	Justification
Plausible Number of people who have the problem	`1B`	There are ~5B[100] people online today, who might have online security concerns
Self-Aware Willing to solve the problem	`0.01`	Have you bought special security software for an online project? Almost no consumer does.
Lucrative Annual allocated budget	`$10`	Consumers don't pay much for things, and globally consumers spend far less on average than Americans on equivalent online goods; arguably this should even be `$1`.
Liquid Frequency of purchase decision	`0.01`	Consumers are rarely in the market.
Eager *(identity)* Attitude towards your company	`0.5`	Unclear without specifics, we'll be generous and assume you can earn trust despite being an unknown brand.

Criteria	Score	Justification
Eager *(comparative)* Competitive differentiation	`0.1`	Consumer-grade security products are undifferentiated.
Enduring Will they still be here a year from now?	`0.5`	Consumers cancel at higher rates than businesses, but at a low price this could be something that is easy to maintain; industry data would be helpful.

Score = `25,000 / 625,000` = **0.04**. This is not a good business model.

But there's hope...

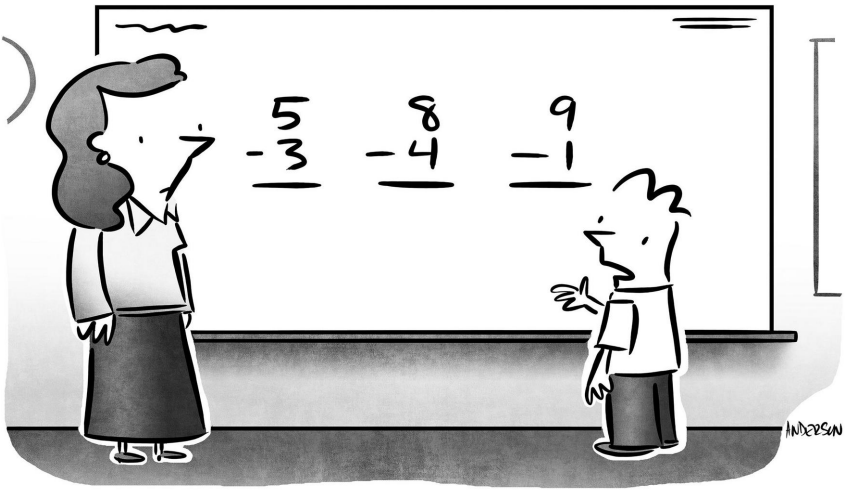
What to do with a negative answer

We don't have to give up just yet.

What if the security company targets high-net-worth individuals instead of "everyone?" The "number of people" would fall to `1,000,000`, but willingness to solve the problem might rise to `1.0`, and budget certainly rises to `$100` if not `$1000`. The orders of magnitude can change dramatically, which might reveal a workable model.

What if the security company targets mid-sized businesses? The number of organizations is smaller than the number of consumers (but is still large), willingness to solve is very high (they have security policies and fear of the downside of a security incident), budgets are substantial, and so on.

In general, **targeting a niche often results in a better business model**, because although it reduces the number of target customers, it can increase several other numbers. This is very often the right answer whether your goal is to build a small, profitable, sustainable business (in which you stay in that niche forever) or a large multi-billion-dollar



credit[101]

"This is why no one likes math; it's a branding issue. Everything's a problem."

enterprise (in which the niche is your way to get started, and you expand the target market over time).

> *"I'd rather be wrong than do something wrong."*
>
> *–Larry Ellison*

Or, finally, your business idea might simply not be viable. That is a sad and tough reality to face (p. 657), but it's better to figure that out early, so you can spend your time finding a new idea.

I hope this framework helps you build a winning strategy!

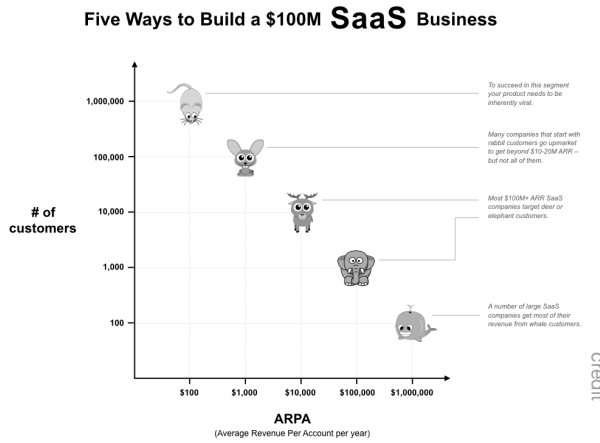


Figure 3

FURTHER READING

- Five ways to build a $100M business,[102] the classic by Christoph Janz that uses five animals to show the different orders-of-magnitude of price versus size of company and quantity of those companies (Figure 3). This is equivalent to two of our rubric lines.
- Pricing determines your business model (p. 515): How orders-of-magnitude in pricing changes your product and target market.
- How to use the Needs Stack (p. 259) to determine both features and benefits that make sense for your customers.
- The factors that cause customers to be willing to pay more (p. 275) for a product.
- Trading off many customers at low price-points versus few at high price-points (p. 1399).
- Why selling to the mid-market[104] can be the "worst of both worlds."
- How small companies can win against Enterprise incumbents (p. 1495).

- Selling to Carol (p. 317): How targeting your one "perfect" customer is the right way to market in general.

Many thanks to Daniel Zarick,[105] *KimSia Sim,*[106] *Matt Wensing,*[107] *Sam Bhagwat,*[108] *Tony Meijer,*[109] *and Willis F Jackson III*[110] *for contributing their insights to early drafts, and to Daniel Veihelmann for the online calculator.*[111]

Chapter 5:
Your customers hate MVPs. Make a SLC instead.

DISILLUSIONED WITH MVP
SIMPLE, LOVABLE, COMPLETE · LIFE AFTER SLC
MASLOW'S PRODUCT HIERARCHY

"What was initially thought to be a simple process is in fact an incredibly complicated, intricate, and complex system that I've codified and organized into a few easy-to-follow rules that are more difficult to implement than you'd think."

credit [12]

DISILLUSIONED WITH MVP

Product teams have been repeating the MVP[113] (Minimum Viable Product) mantra for a decade now, without re-evaluating whether it's the right way to maximize learning while pleasing the customer.

Well, it's not the best system. It's selfish and it hurts customers. We don't build MVPs at WP Engine.

The motivation behind the MVP is still valid:

1. Build something small, because small things are quick and inexpensive to test.
2. Get it into the market quickly, because real learning occurs only when real customers are using a real product.
3. Trash it or hard-pivot if it's a failure (p. 1261), or invest if it's a seedling with potential.

MVPs are great for startups and product teams because they maximize so-called "validated learning" as quickly as possible. And while customer interviews (p. 239) are useful, you learn new things when a customer actually uses the product. But MVPs are a selfish act.

The problem is: Customers hate MVPs. Startups are encouraged by the great Reid Hoffman[114] to "launch early enough that you're embarrassed by your v1.0 release." But no customer wants to use an unfinished product that the creators are embarrassed by. Customers want *great* products they can use *now*.

MVPs are too M and rarely V. Customers see that, and hate it. It might be great for the product team, but it's bad for customers. And ultimately, what's bad for customers is bad for the company.

Fortunately, there's a better way to build and validate products. The insight comes from honoring the utility of MVPs (listed above) while giving just as much consideration to the customer's experience.

SLC: SIMPLE, LOVABLE, COMPLETE

In order for the product to be small and delivered quickly, it has to be **simple**. Customers accept simple products every day. Even if it doesn't do everything needed, as long as the product never *claimed* to do more than it does, customers are forgiving. For example, it was okay that early versions Google Docs had only 3% of the features of Microsoft Word, because Docs did a great job at what it was primarily designed for, which is simplicity and real-time collaboration.

Google Docs was simple, but also **complete**. This is decidedly different from the classic MVP, which by definition isn't complete (in fact, it's "embarrassing"). "Simple" is good, "incomplete" is not. The customer should have a genuine desire to use the product, *as-is*. Not because it's version 0.1 of something complex, but because it's version 1.0 of something simple.

It is not contradictory for products to be simple as well as complete. Examples include the first versions of WhatsApp, Snapchat, Stripe, Twilio, Twitter, and Slack. Some of those later expanded to add complexity (Snapchat, Stripe, Slack), whereas some kept it simple as a permanent value (Twitter, WhatsApp). Virgin Air and Southwest Airlines both started with just one route. Southwest Airlines is the most profitable US airline in history. Small, but complete.

The final ingredient, and the one most unlike MVP, is for the product to be **lovable**. People have to *want* to use it. Products that do less but are loved, are more successful than products which have more features and are disliked. The original, very-low-feature, very-highly-

loved, hyper-successful early versions of all the products listed in the previous paragraph are examples. The Darwinian success loop[115] of a product is a function of love, not of features.

There are many ways to generate love. "Minimum" and "viable" are definitely *not* among those ways. The current-in-vogue way is through design: Elegant UX combined with delightful UI. But there are other ways. The attitude and culture of the company itself can generate love, such as Buffer's blog[116] with its delightfully shocking transparency (including open salaries and corporate metrics) or MeetEdgar's blog[117] genuinely helping entrepreneurs or HubSpot's blog[118] which early on was at least as instrumental to their customers' success as the actual product. Another way is through a deep connection to the psyche of customers, like Heroku who broke with marketing tradition by filling the homepage with command-line examples instead of benefit-statements, thereby connecting instantly with their geeky target customer (Figure 1).

Read about "WTP" (p. 275) for many more examples of how to generate love.

Figure 1

From this reasoning, years ago I named what I believe is the correct alternative to the MVP: Simple, Lovable and Complete (**SLC**). We pronounce it "Slick." As in: "What's the 'Slick' version of your idea?"

SLC Summary

Simple, because complex things can't be built quickly, and you must ship quickly so you can learn quickly so you can create the right product before you run out of money and willpower.

Lovable, because crappy products are insulting, and you didn't start this company to make crappy products. The love overpowers the fact that the product is buggy and feature-poor. There are many wonderful, powerful, competitively-defensible forms of "Love." (p. 275) Pick some.

Complete, because products are supposed to accomplish a job. Customers want to use a v1 of something simple, not v0.1 of something broken.

LIFE AFTER SLC

Another benefit of SLC becomes apparent when you consider the next version of the product.

A SLC product does not require on-going development. It is possible that v1 should evolve for years into a v4, but you also have the option of *not* investing further, yet the product still creates value. An

MVP that never gets additional investment is just a bad product. A SLC that never gets additional investment is a good, if modest product.

Many of the most successful software products in the world started as SLC, then grew in complexity, including examples we've already given (Google Docs and Snapchat).

The first iteration of Snapchat was a screen where tapping anywhere took a picture, that you could then send to someone else, at which time it disappeared. No video, no filters, no social networking, no commenting and no storage—Simple, yet Lovable and Complete, as evidenced by its rapid adoption. The insight of "no storage" was critical, but many people have theorized that the simplicity of the interface was also critical. The very fact that it was simple, while not sacrificing love-ability or completeness, caused its success.

Later they added lots of stuff—video, filters, timelines, "stories", even video cameras inside sunglasses. It's OK for products to become complex. Starting out SLC does not preclude becoming complex later.

WhatsApp was similar; it started with just a status message. Not a "post", not a "chat", no "timelines", no "history". Just "What's up?", hence the name of the app. They found people abusing the system to communicate with each other without paying for SMS messages, so they added chat. Dropbox started with just one folder that would (eventually!) sync across devices. Twitter had only the 140-character messages; things like replies and re-tweets were invented by users, implemented by convention, and only later folded back into the platform as built-in features. The examples go on and on.

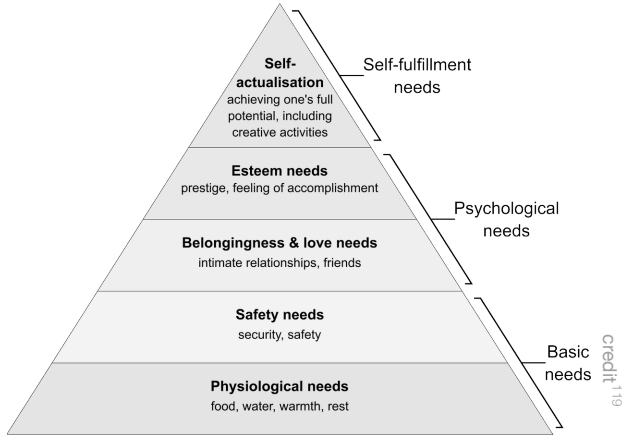


Figure 2: Maslow's Hierarchy of Needs

SLC BREAKS MASLOW'S PRODUCT HIERARCHY OF NEEDS

People erroneously believe that product development should work like Abraham Maslow's Hierarchy of Needs (Figure 2).

His insight is that you cannot achieve higher levels if you haven't at least satisfied the lower levels: If you don't know where your next meal is coming from, you're not able to "creatively discover your true self."

The framework is wrong, though this hasn't stopped people from writing books and blog posts about it. "At the time of its original publication in 1943, there was no empirical evidence to support the theory." "In a 1976 review of Maslow's hierarchy of needs, little evidence was found for the specific ranking of needs that Maslow described or for the existence of a definite hierarchy at all." *(From Wikipedia,*[120] *with its own references.)*

It's wrong for Product also, but we act like it's true.

If we roughly parallel Maslow, the Product Hierarchy looks something like (Figure 3), defining the levels as:

Meaningful
Identity, belonging, higher purpose. "Larger than myself." Impact. Legacy (p. 561).

Delightful
Exceptionally wonderful to use, eliciting a strong positive emotion. It goes beyond being visually appealing or sensibly designed. It is at least extreme; it is likely surprising, because so few products delight us, especially in the business world. Perhaps a low bar makes delight easier to achieve, if we try.

Easy to use
Effortless and intuitive. Makes tech support obsolete.

Reliable
Always works as promised.

Useful
Fulfills a need; solves a problem.

Following Maslow, we might say that if a product isn't Useful then it doesn't matter if it works all the time (Reliable) or is pretty to look at (Delightful), therefore "being useful" is the mandatory first "rung" of the ladder. No reason to do anything else, if you're not useful.

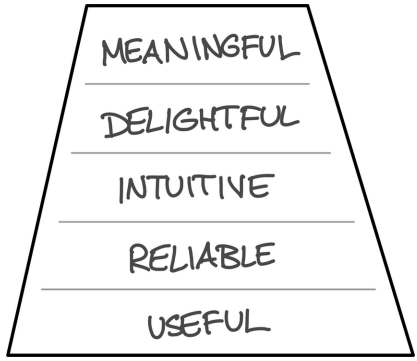


Figure 3: Pseudo-Maslow Product Hierarchy

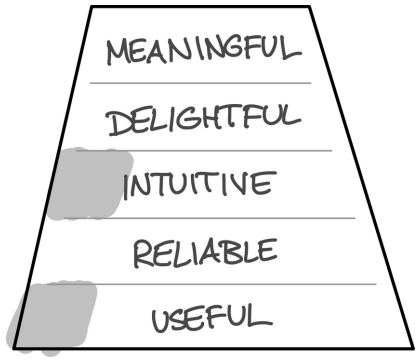


Figure 4: **The typical MVP** is minimally useful only, perhaps also being easy to try.

Indeed, that's what traditional MVPs do: be minimally useful, disregarding all other levels as irrelevant until the first level is satisfied (Figure 4).

This MVP attitude is further justified by projecting its future. We can map the behavior of mature products, especially in so-called "Enterprise Software" where the person who chooses to buy it isn't the person who is forced to use it, and therefore the bottom of the pyramid is valued and the top is not (Figure 5).

The SLC attitude is different. It agrees that we can't fill much of any layer, because we need to ship it quickly and start getting feedback. But it emphasizes different layers (Figure 6).

An SLC product evolves, already-happy customers are rewarded with additional features (Figure 7).

Perhaps a better way to look at it, is that SLC is both Delightful *and Useful* on day one, albeit with a *scope* of "Useful" that is small enough to be "Complete" (Figure 8).

Finally, a product that from inception is trying to "delight customers" is one that might actually deliver on the top of the pyramid: Meaning, personal identity, a higher purpose. You see this in products

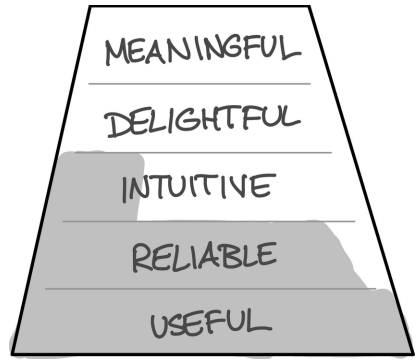


Figure 5: **Mature products** maximize the utility layers; do their users love it, or are they making an internal case for why it should be replaced with a more pleasant competitor?

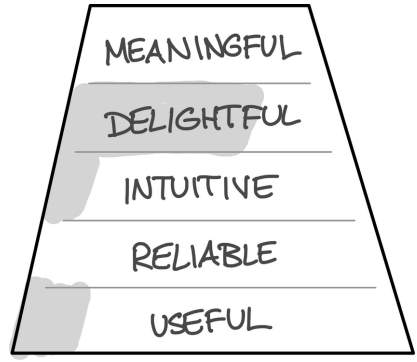


Figure 6: **SLC initially** uses Delight to win the hearts of customers even though the product isn't as useful or reliable as it will eventually be.

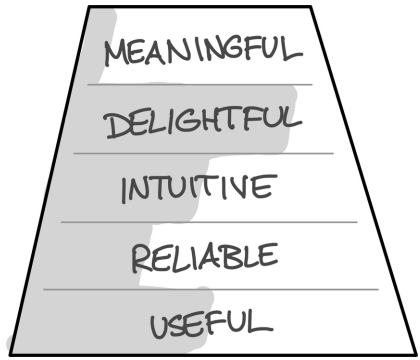


Figure 7: *SLC evolves* in a fundamentally different way, creating differentiation beyond feature-bullet-points.

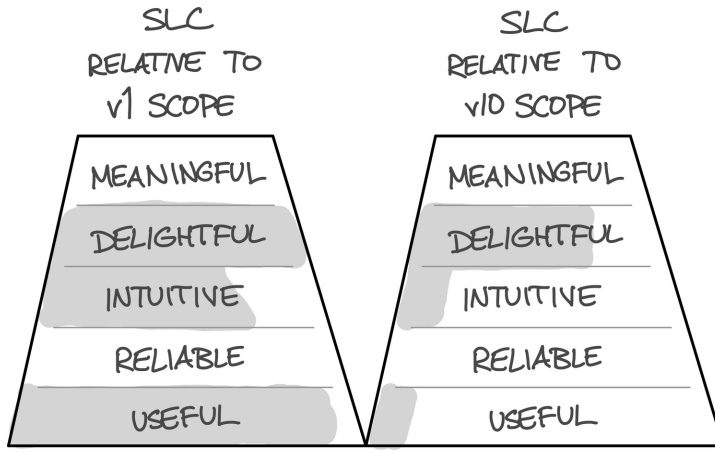


Figure 8: **SLC as complete** according to a narrow scope (left), but lacking features relative to the scope of a mature product (right).

where people tie their identity to owning or using the product, not just in consumer brands where this is obvious, but in the way that people love Linear because it honors the developer (instead of the project manager), or the way they love Basecamp because they support the culture and attitudes of 37signals, or the way they whip out Moleskine notebooks because it connects them to a romantic ideal of the brooding writer, or proudly wear Patagonia gear as a badge of their environmental consciousness, or the way they argue over Vim vs Emacs as a badge of geekery.

Almost no company cares about creating meaning for their customers. Here's how I know: What metric are they tracking (p. 645)—let alone optimizing for—that measures meaning? If none, then it's out of sight, out of mind. Clearly, you can build large and even great companies like this; most do. But if you did care about that, SLC is the right way to start the adventure.

Why don't more products prioritize delight? Utility is more obvious. Utility is what the customer's budget is allocated to obtain. Delight requires insight and great design, not back-end optimization and a keen mathematical mind. It is a rare person who possess all of these abilities, or even values them in others. It is a *decision* to prioritize Delight over Utility; it is easier not to.

The "pyramid" is useful for mapping out where you're going to spend your time, but we need not traverse it from bottom to top. Products that prioritize delight win over products that don't.

It's really another way of prioritizing the customer—the human being, not just their "job to be done."

With SLC, the outcomes are better and your options for next steps are better. It might fail; both SLCs and MVPs sometimes produce that result because you're running an experiment (p. 1261). But if a SLC succeeds, you've already delivered real value to customers and you

have multiple futures available to you, none of which are urgent. You could build a v2, and because you're already generating value, you have more time to decide what that should look like. You could even query existing customers to determine exactly what v2 should entail, instead of a set of alpha-testers who just want to know "when are you going to fix this broken thing?"

Or, you can decide not to work on it. Not every product has to become complex. Not every product needs new major versions every two quarters. Some things can just remain simple, lovable, and complete.

Ask your customers. They'll agree.

Many thanks to Devan Stormont,[121] *Kathy Qian,*[122] *and Khurrum Mahmood*[123] *for feedback on this article.*

Chapter 6:
The Elephant in the room: The myth of exponential hypergrowth

DISPELLING "EXPONENTIAL"
HYPERGROWTH IS QUADRATIC · MARKETING
VIRALITY · ACTIONABLE CONCLUSIONS

A startup is growing fast, the journalists marveling at its "meteoric rise." But don't meteors fall?

Inevitably it is breathlessly inducted into the class of "hypergrowth" companies that are "growing exponentially." Especially when the product is "viral." After all, if every person brings three friends, and each of those brings another three, is that not exponential?

But "exponential" is an incorrect characterization, as we'll see in real-world data, even for hypergrowth, "viral" companies like Facebook and Slack.

This article suggests an alternate model for how fast-growing companies actually grow. Understanding the model is useful not only for predicting growth, but because understanding the foundational drivers of growth allows us to take smarter actions to create growth in our own companies.

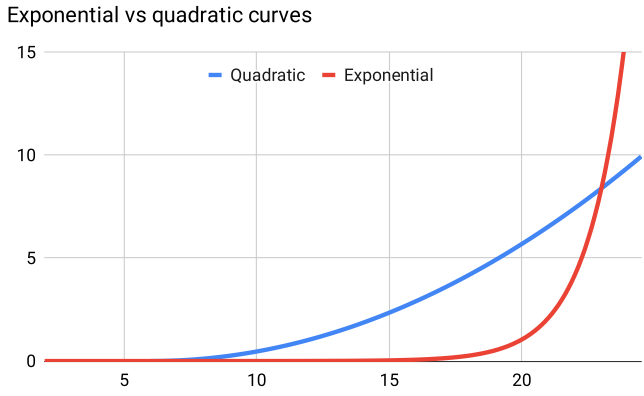


Figure 1

DISPELLING "EXPONENTIAL"

To evaluate whether hypergrowth is properly described as "exponential," let's recall what that word means. Here's an exponential curve (like $y = 2^x$), compared to a quadratic one (like $y = x^2$) (Figure 1).

In exponential growth, values grow by a *multiple*. For example: In year 1 you grow 10, in year 2 by 100, in year 3 by 1000—each time the amount of growth is *multiplied* by ten. The compounding effect of multiplication causes the numbers to grow slowly initially, then skyrocket. The compounding effect gets journalists and VCs justifiably excited.

> *"Compound interest is the most powerful force in the universe."*
>
> *–Albert Einstein*

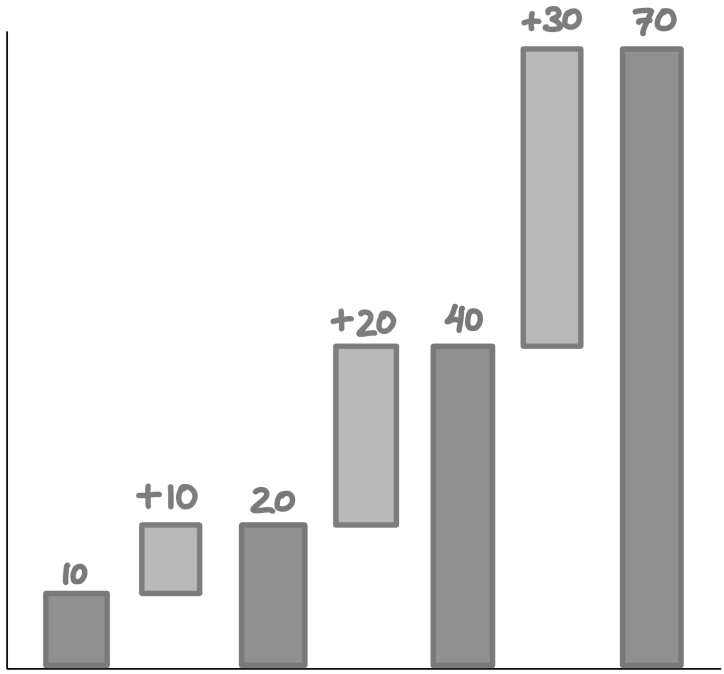


Figure 2: Successive values (in blue) are increasing more and more (in green). The green differences are increasing linearly: 10, 20, 30.

In quadratic growth, values grow by a *adding* a constant amount more each time-interval, rather than *multiplying* a constant amount more each time-interval. In the same example, growing in year 1 by 10, then in year 2 by 20, in year 3 by 30 (Figure 2).

Growth is still *accelerating*, so the blue curve slopes upwards, but gently compared to exponential growth.

With these patterns in mind, let's examine real-world data, and see whether "exponential" is the right model.

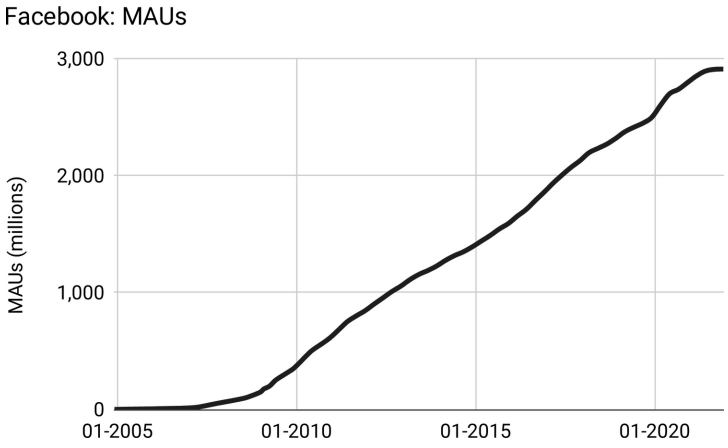


Figure 3: Essentially linear for nearly twenty years, only exponential in the first four years.

Facebook is the definition of hypergrowth—getting to $50B in revenue faster[124] than any company in history. The product is "viral"—friends bring other friends—which theoretically leads to "exponential growth." But Facebook didn't grow exponentially in the number of monthly active users (Figure 3).

Slack was the fastest-growing enterprise software company ever,[125] going from $0 to $10M ARR in their first 10 months, and 0 to 10,000,000 active users in just five years. It's also a "viral" product—organizations invite their members, who then create their own Slack-groups and invite others. So surely Slack has exponential growth? (Figure 4)

If you compare Slack's growth with **Microsoft Team**'s growth, do you still think Slack's growth is "exponential?" (Figure 5)

Dropbox was another "hypergrowth" company, achieving 100,000,000 registered users five years after being founded in 2007,

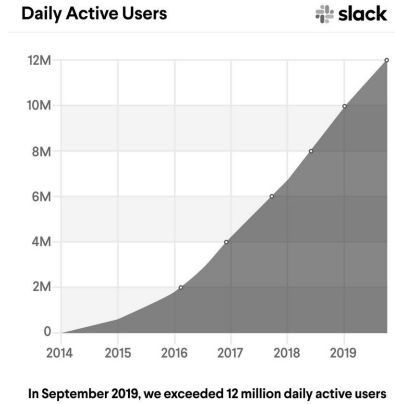


Figure 4: Slack's own data[126] shows initial quadratic growth, followed by years of linear growth.

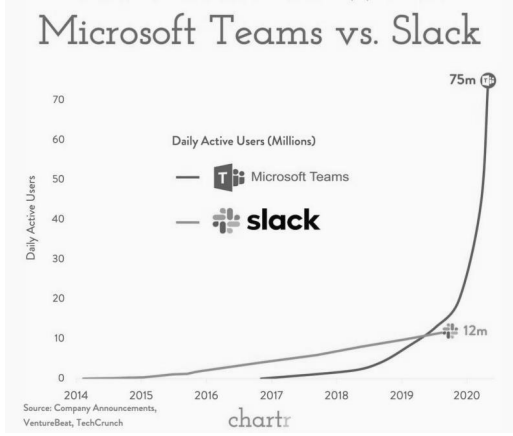


Figure 5

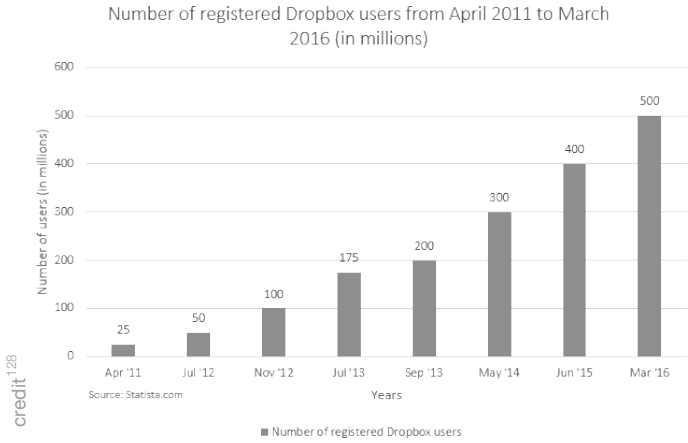


Figure 6: Early in life, Dropbox registered users grows non-exponentially, nearly exactly 100M per year

but it wasn't exponential, neither in freemium users nor in revenue, early nor later in life (Figure 6) (Figure 7).

Trello grew fast too, getting to 10,000,000 registered users in three years. But not exponentially (Figure 8).

Lyft grew in part due to "network effects" according to their S1,[131] but this chart they presented shows that active rider growth isn't exponential (Figure 9).

Hubspot's revenue curve is astonishingly consistent, despite hitting multiple inflection points* in their business (Figure 10).

Analyzing this last example, we arrive at a new, non-exponential model.

* e.g. launching new business models like selling through agencies instead of only directly, launching new product lines like sales CRM on top of marketing automation tools, and scaling the number of customers and employees by 10x

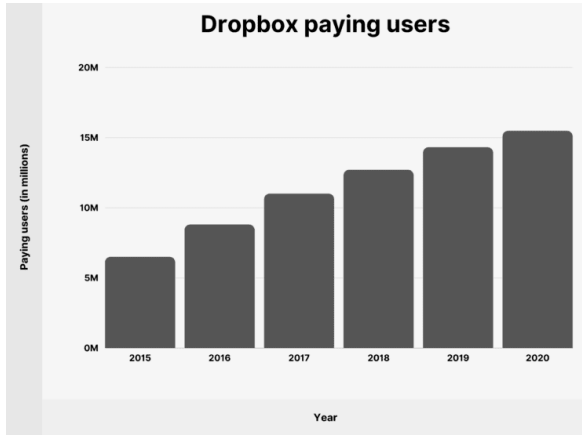


Figure 7: Later in life, Dropbox revenue grows linearly, and slows down

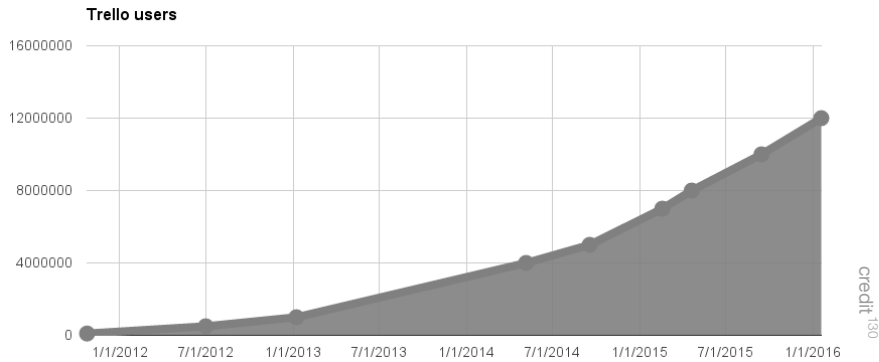


Figure 8

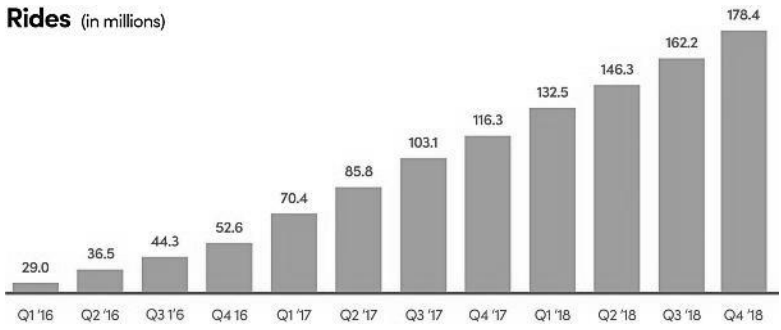


Figure 9

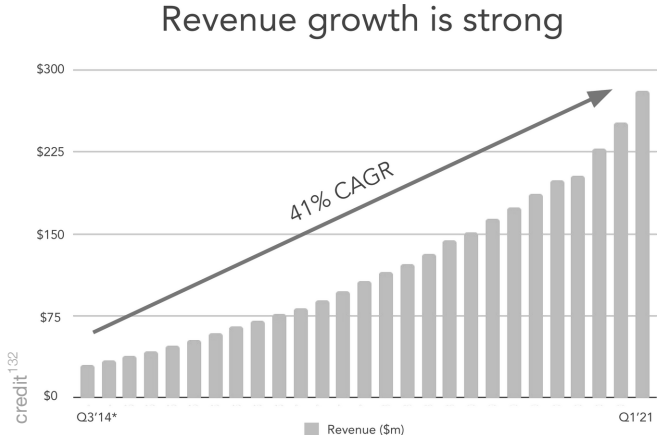


Figure 10

HYPERGROWTH IS QUADRATIC

The language we use can determine[133] the thoughts we have.

The Hubspot slide says "41% CAGR." "CAGR" means annualized growth rate. They're saying that if you start with the first number on the slide, then from there plot growing 41% per each year, compounding each year upon the previous, for seven years, you would arrive at the last number on the slide. This is exactly the definition of "exponential"—multiplying by a number repeatedly. In general when you use "CAGR" or "percentage growth" as a metric, you are implicitly saying "This is an exponential process."

But Hubspot didn't grow by 41% every year; in this time frame, it started at 60% and ended around 30% (Figure 11).

If instead we examine growth in absolute dollars, rather than in percent, a pattern emerges. In the first set of four quarters on this report, they grew $17M. The next set grew $23M. Then $28M. Then $34M. Each year $5-7M more than the previous. This is the definition of a quadratic—*adding* an amount that increases by a constant amount each period, not *multiplying*.

Charting these year-over-year revenue differences *in absolute dollars* rather than *in percent*, it's clear that indeed the changes were al-

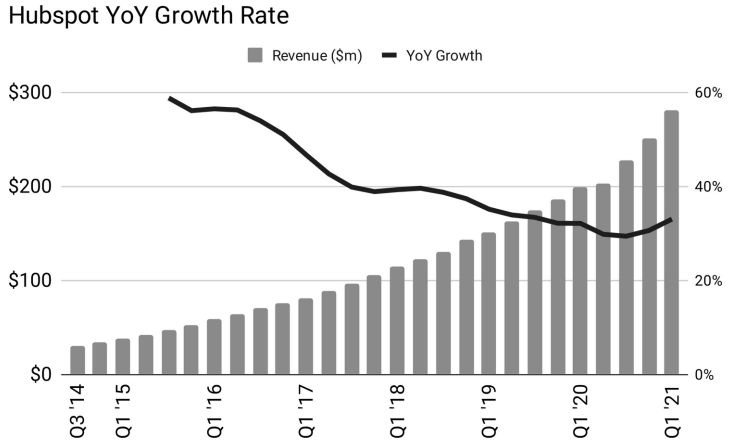


Figure 11: Exponential curves have a *constant* year-over-year growth rate, therefore this is *not* exponential growth.

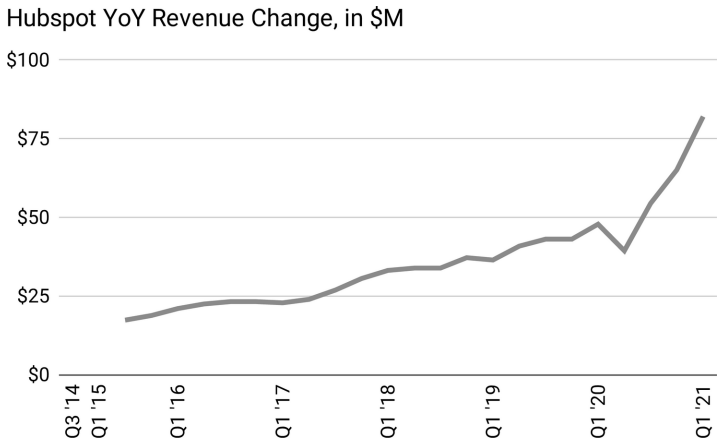


Figure 12

most completely linear for years, then suddenly changed in 2020* to a new (but still linear) rate (Figure 12).

It is therefore mathematically inevitable that plotting a quadratic curve (rather than exponential) on top of Hubspot's revenue data will be a perfect fit (Figure 13).

My thesis is that **High-growth companies grow quadratically, not exponentially.****

The consequence of this conclusion is important for operators and analysis and investors. These are all people trying to understand—and possibly change—growth drivers. Getting the right language, and the right model, will lead to right analysis, and right action.

---

* Coinciding with the launch of a new product: Hubspot CMS Hub.

** My guess is low-growth companies are similar, but data are more easily available for the runaway-growth companies who publicly flaunt their success.

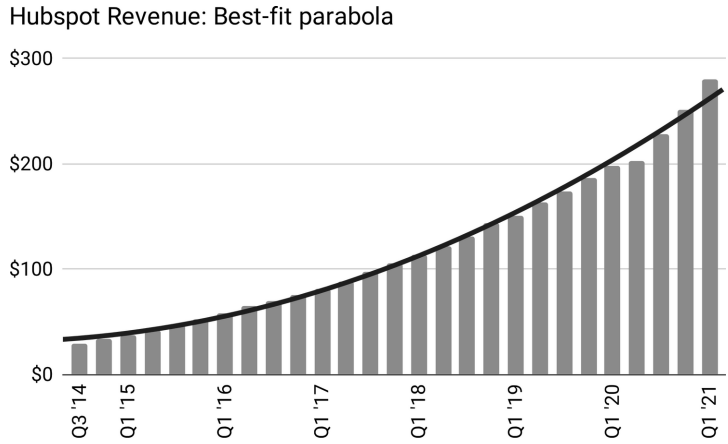


Figure 13: When you said "best-fit," you weren't kidding!

WHY MARKETING-DRIVEN PRODUCTS GROW QUADRATICALLY: A FIRST-PRINCIPLES EXPLANATION

It's not enough to draw best-fit lines on top of PowerPoint slides. We have to explain *why* this model makes sense, which in turn will create a better understanding of the growth drivers in our own companies.

We've been taking a macro view of growth, looking at multi-year trajectories. Now we'll peer into the microscope instead of the telescope, and consider how growth arises from a single marketing campaign.

The life of a marketing campaign

In my experience, marketing campaigns follow this pattern (Figure 14).

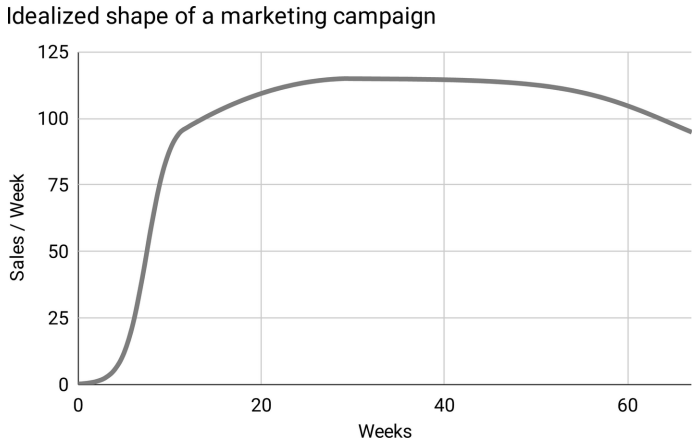


Figure 14

At the foot of the curve, we've launched a new campaign, but it's ineffective; we haven't figured out the best design and messaging and calls-to-action for this new medium and audience. Sometimes we never figure it out, and abandon the effort.*

But in the case that we unlock the secret of efficacy, the campaign rapidly reaches a natural level of contribution; in this example, a number of "sales per week." The specific level depends on many things: ad inventory, our budget, audience-receptivity, and the consonance between the audience and our target market.

Next we enter the optimization phase. We A/B Test our way to incrementally better results. Also we enjoy the result of multiple exposures—most people need to see the ad more than once before they act.

Finally we enter a phase of decline. There are various causes, all instructive:

* It's hard to distinguish (a) our failure to build effective copy and conversion funnels from (b) channels that are fundamentally a bad fit for our market or product. This uncertainty, together with the rapid evolution of digital marketing, suggests that we should retry campaigns in previously-failed channels every few years.

- **The audience saturates.** Everyone in the channel has seen the ad more times than is required to act; it's now falling on deaf ears. Even if the audience is growing, the number of new people is small compared to the number of people that were new-to-us when we began the campaign.
- **The channel declines.** A media site that was popular loses readers through over-monetization. An event that was well-attended loses favor. A newsletter that was frequent and insightful becomes less frequent or other writers take over. A podcast moves to a closed platform and loses many listeners.
- **The auction becomes uneconomical.** For auction-based systems like Google and Facebook advertisements, or other zero-sum (p. 295) programs like affiliates or limited-inventory spots on newsletters or podcasts, the winner is the one who will pay the most. What is cost-effective for one bidder will be laughably over-priced for another, due to better conversion rates, higher revenue per customer, higher profitability per customer, or due to categorization as a "loss leader" or other way of ascribing value beyond immediate pay-back.

This curve leads to actionable ideas for managing marketing (given at the end of this article), but also forms my central thesis about how all sorts of growth works at companies. So I'm giving it a name (Figure 15).

How the idealized marketing campaign converts to growth

The model above shows the number of sales *per week* the campaign contributes. To understand how this looks in terms of revenue growth, let us suppose a simple business model in which all sales are for a recurring revenue product generating $10 per month, with a 1% per

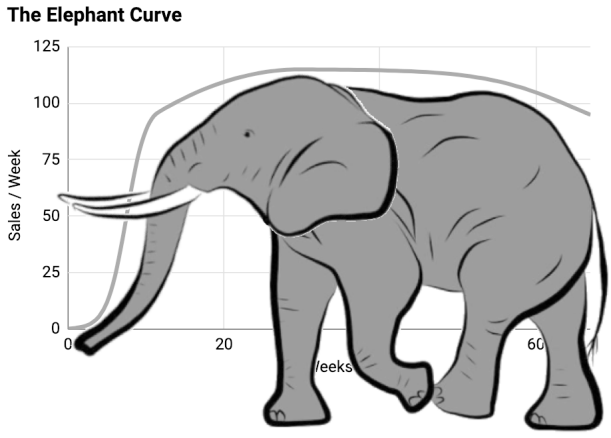


Figure 15

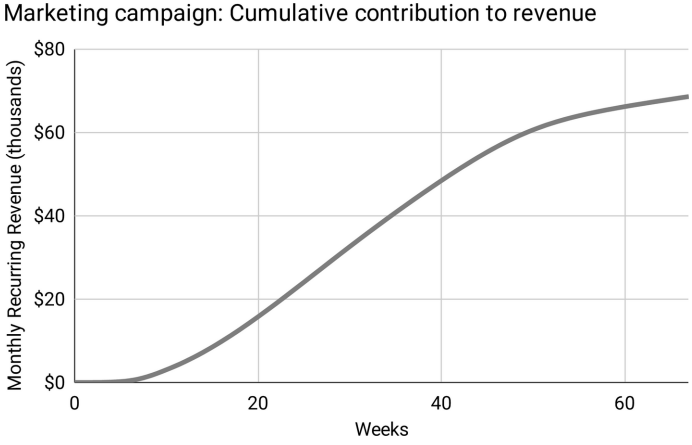


Figure 16

month cancellation rate. Revenue grows over time in a certain way (Figure 16).

Growth initially accelerates as the campaign is solved, then grows roughly linearly as the campaign is optimized, and then starts sagging (although still growing!) as the campaign declines, and as the now-sizable customer base produces a non-trivial number of cancellations.

The layer-cake of quadratic growth

Marketing departments don't stop at a single campaign. They add new ones. Some are bigger than others, some can be optimized more than others, some decline sooner than others, some decline more precipitously than others.

So, let's model that: A variety of Elephant Curves, with differing parameters, beginning at different times, stacking the revenue-contribution of each to arrive at overall revenue growth for the company (Figure 17).

Scan your eye across the top of this kaleidoscopic cake, and you trace a wavy quadratic. This makes sense mathematically, because each campaign is essentially linear after it gets going, even if it sags during

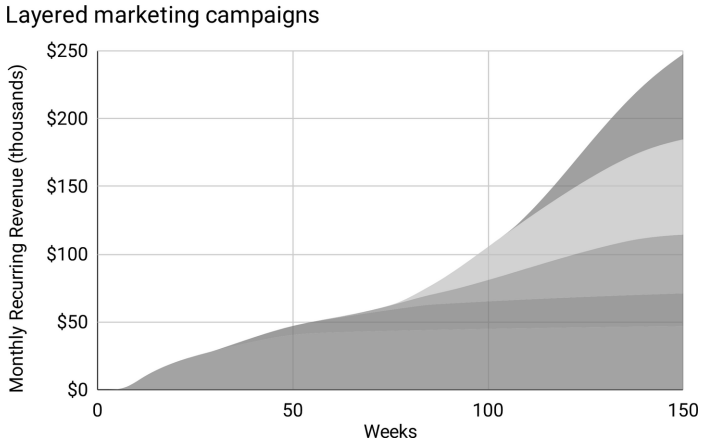


Figure 17: Layered campaigns create a "wavy quadratic."

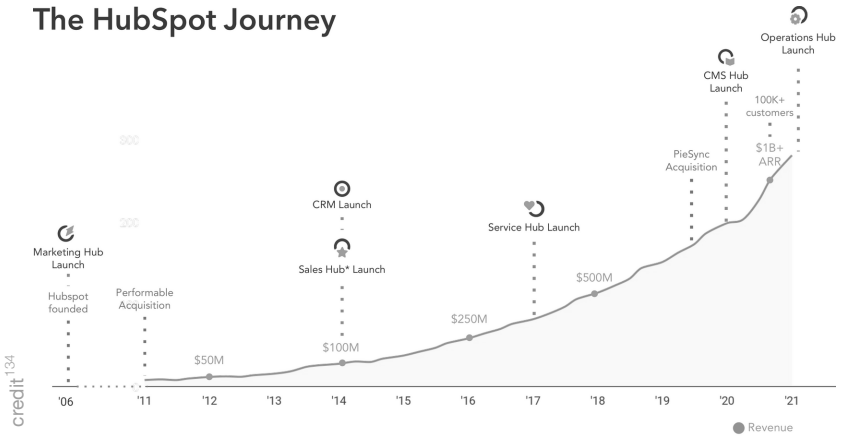


Figure 18

decline. "Adding more linear things over time" is the definition of a quadratic.

The reason it's "wavy," is that when we unlock a new campaign we get a burst of growth. Do real-life revenue curves exhibit this waviness? Maybe so; here's another slide from the Hubspot deck (Figure 18).

Hubspot didn't just add new marketing channels, however, but also layered on new geographies and new products. Do those activities have the same effect as marketing campaigns?

Multiple product lines at marketing-driven companies: Still quadratic

So far we've assumed a single product, driven by marketing campaigns. High-growth companies who want to continue growing quickly after their first product reaches scale, must launch new products into new markets.

Is the Elephant Curve also the shape of an entire product line? After all, products often have an initial slow-growth period (because

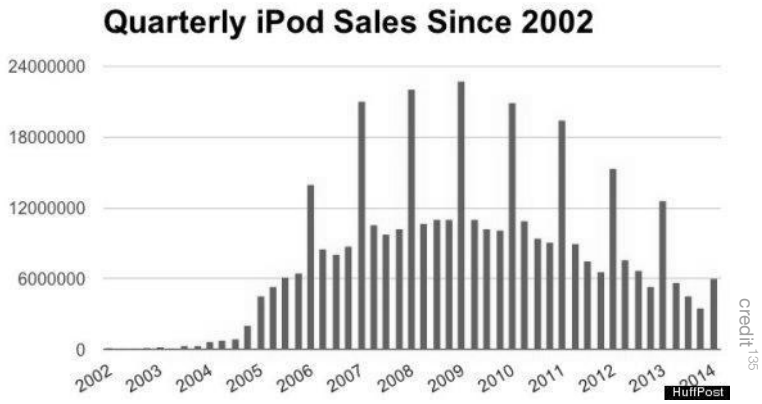


Figure 19

only cutting-edge early adopters are eager to pay to "be first" with bugs and missing features), followed by a faster expansion period, then reach some sort of natural ceiling, and possibly enter a period of decline (as the market evolves or competition overwhelms).

Indeed, this is what we see with many products, especially those that are marketing-driven, and without recurring-revenue. iPod sales, for example, are a perfect match (Figure 19).

It should therefore be unsurprising when we look at the overall revenue chart for Apple, and once again see quadratic growth on the top-line, admittedly with a special one-time bump for the unprecedented* success of the iPhone (Figure 20).

Each product is in a different phase of its lifecycle: The iPod declined to zero, the iPad is still declining; Macs are teetering but essentially flat; iPhones and software services are still increasing.

* It is rare for a second product to dramatically outpace the first; even juggernauts like Google, Amazon, and Facebook never achieved that.

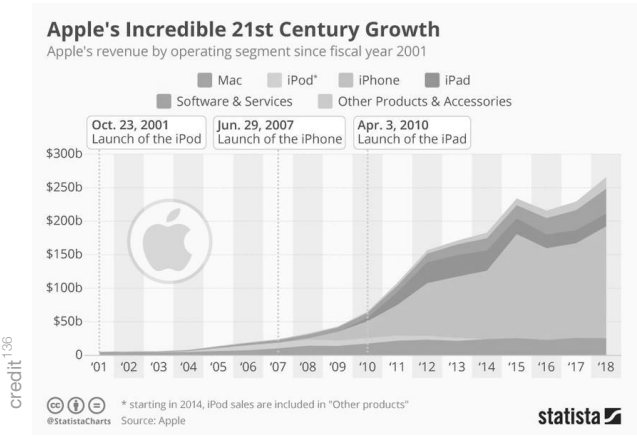


Figure 20

The quadratic explanation for "growth decay"

It's well-known that growth—as a percentage—naturally declines with scale, even when there's nothing wrong with the company.

This law of nature has been given a name: Growth Decay (or sometimes Growth Persistence). Because of the traditional insistence of talking about growth as a percentage, the concept is articulated this way: If a company grew X% last year, it's likely to grow a bit less than X% this year. With this formulation, the question becomes: How much less?

The data give us the answer of 85%, although with $R^2 = 0.51$, this is a tendency but far from a law (Figure 21).

With our new appreciation that growth isn't exponential, and therefore "percentage" might be the wrong way to characterize growth, we could ask what curve would best model the idea of Growth Decay? Specifically, let's plot revenue for an initially-fast-growing company that is subject to the principle of Growth Decay (Figure 22).

The first sixteen years of the curve is quadratic. While mathematically not identical, the best-fit quadratic curve* has a staggering $R^2 = 0.999$.

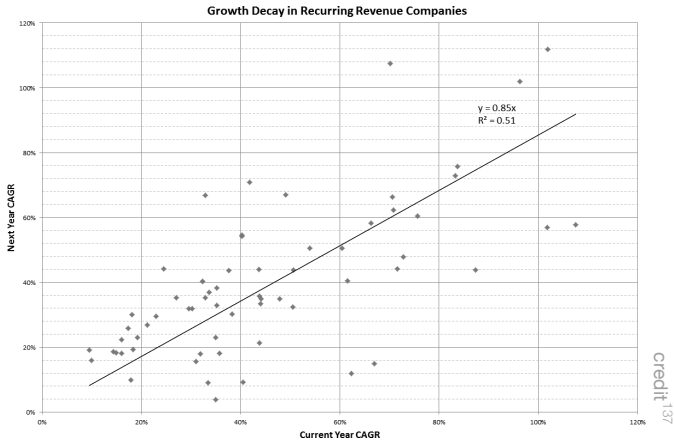


Figure 21

This is yet another signal that quadratic growth is the correct model.

BEYOND MARKETING CAMPAIGNS: "VIRAL" AND OTHER FORMS OF "EXPONENTIAL" GROWTH

But some products really do grow exponentially. In theory.

In theory, theory and practice are the same. In practice, they're not.**

---

* $y = 13.6x^2 - 43.4x + 34.6$ when fit perfectly; interestingly this is very close to $y = 13.6(x - 1.595)^2$, showing how simple the curve really is.

** This phrase attributed to Benjamin Brewster.

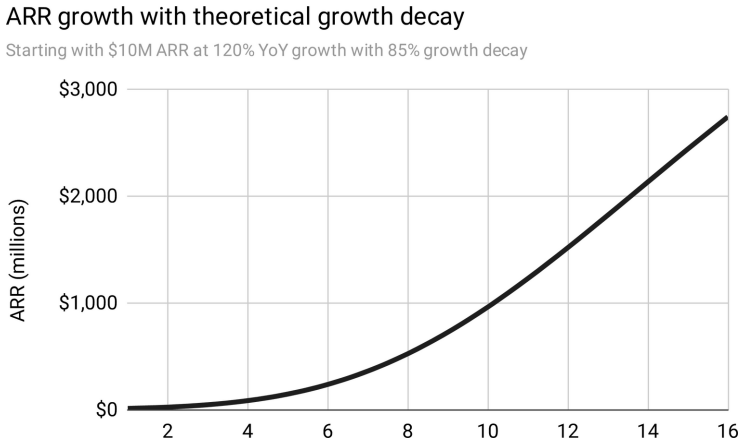


Figure 22

Some products don't grow proportionally with marketing and sales, but instead self-propel with a mechanism that theoretically ought to be exponential. There are at least three ways for this to happen:

Type 1: Virality
When each user invites on average another a users, then each of *those* a new users bring in *another* a new users, so we end up with a^2 more. Then each of *those* brings in another a which yields a^3. Then a^4 and so on; this is the definition of exponential growth. Biological viruses grow exponentially for a similar reason, justifying the label.

Examples: social media, chat clients, peer-to-peer payment platforms, massively-multi-player games, fantasy sports leagues.

Type 2: Word-of-Mouth
All products have some word-of-mouth component, but here we're referring to products that are *primarily* driven this way; this creates a growth process that is similar to viral. Typically the mechanism of "telling others" is built into the product, rather than bolted on by marketing or generated by goodwill. The difference between "word-of-mouth" and "viral," is that viral products are *unusable* unless you invite others to become users (thus exponential growth is enforced) whereas word-of-

mouth products *encourage* sharing. Thus chat clients are viral because without inviting others you can't chat, whereas Wordle* was word-of-mouth, because you play the game alone, but are encouraged to share results on Twitter, which in turn brings in new users.

Examples: gamified products that generate significant sharing (self-improvement, game-results), consumer-to-consumer marketplaces where being a buyer plants the idea of becoming a seller (eBay, Airbnb, Uber); organizations with a cause that creates on-going buzz (brazenly unique cultures, a passionate higher purpose, something people feel is linked to their personal identity).

Type 3: Hot Trend
Products that "everyone" (in some well-defined market) is going to buy. For smartphones, that might be half the population of the world. For internet search, that might be 100% of the online world. For backend management systems for large hospital chains, that could be 1000 potential customers. These products hit "tipping points" where "suddenly everyone buys it." Even if, like internet search, the product has no explicitly viral nor word-of-mouth component—when you search on Google, you don't "invite friends" to also search on Google—the ubiquity and inevitability of the trend leads to an explosion of users.

Examples: word-processing, spreadsheets, broadband internet, the smartphone, the shifts to cloud computing and online shopping, major media delivery platforms of radio, TV, DVD, and video streaming.

Logistic growth: Nearly the right model for virality

Products cannot grow forever, for the obvious reason that markets are finite. The Facebook virus spread to billions of people, but not infinite. Smartphones have been purchased by billions of people, but not infinite.

* Wordle exploded[138] from 90 players in November 2021, to 300,000 in December, to 2,000,000 in January, when it was bought[139] by the New York Times.

Therefore, even if "exponential" is the correct model for the core growth mechanism of the product, it nevertheless cannot continue growing exponentially because it runs out of market. Furthermore, markets tend to have so-called "low-hanging fruit"—customers who are more eager to buy—so although the virus spreads exponentially through these easy-pickings, it runs into the majority of people who will buy, but maybe later, maybe after more of their friends or competitors are using it, maybe if it's less expensive, maybe once it has more features, maybe once it supports integration with specific other software, and all manner of other excuses. The virus has more trouble infecting these high-strung fruits, so growth slows.

This suggests a curve that starts exponentially, but then slows as it runs into the soft back-pressure of more demanding customers, and finally flattens out completely as it runs into the hard limit of the size of the addressable market.

Biologists have already done the work for us, because this is the correct model not just for viral products, but biological viruses infecting a population—akin to product types 1 and 2 above. Intriguingly, this is also the correct model* for the diffusion of a gas across a membrane—akin to product type 3. The mathematical model for all of these processes is the logistic curve:

The logistic curve *is* exponential in the early days when it is far away from its natural limit. As the product (or gas or virus) gets to around 25% market penetration (or infections or saturation), the curve flattens into linear growth, in a tension between the exponential force of growth, countered by fewer and more demanding remaining targets. Finally it levels out at what is called the "carrying capacity"—the fully-saturated market.

* The similarity is that in both cases you have a sudden demand that enters into a new space, but which slows and eventually stops as the new space becomes saturated.

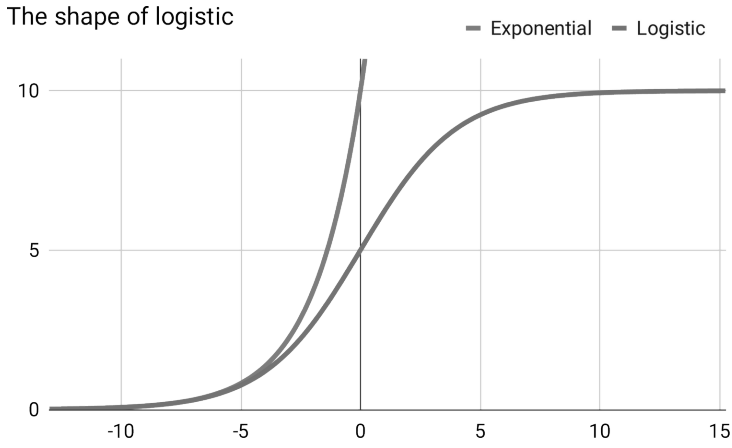


Figure 23

The logistic curve is evident in the real world, in all three product types: (Figure 24) (Figure 25) (Figure 26) (Figure 27) (Figure 28) (Figure 29) (Figure 30)

Stacking logistic growth: The quadratic reappears

Marketing-driven products demonstrated quadratic growth, especially once Elephant-shaped campaigns and products were stacked. How does this differ with logistic growth?

As already pointed out, logistic growth is similar to the Elephant Curve. The "high growth" portion of a marketing campaign might in fact be logistic; a product might extend that period into years rather than weeks, and the absolute magnitude of the result might be many times larger.

If this idea is correct, we ought to see viral-like products exhibit a similar curve to the iPod curve—i.e. a product with initially-exponential growth, then a flattening, perhaps with some small growth, then on a

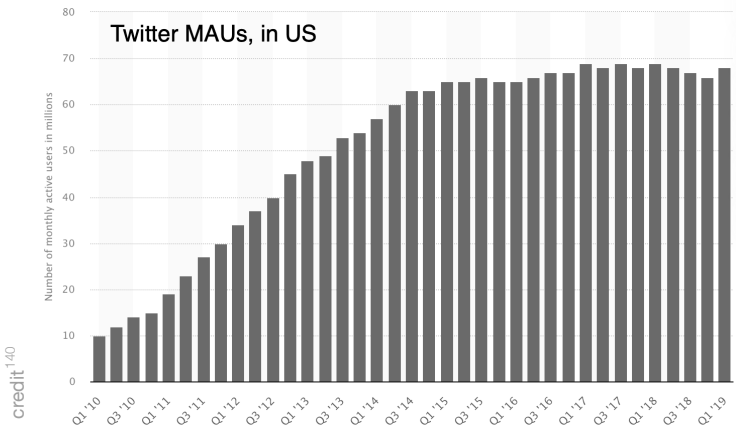


Figure 24: Twitter is a type 1 “Viral” product that follows the logistic model

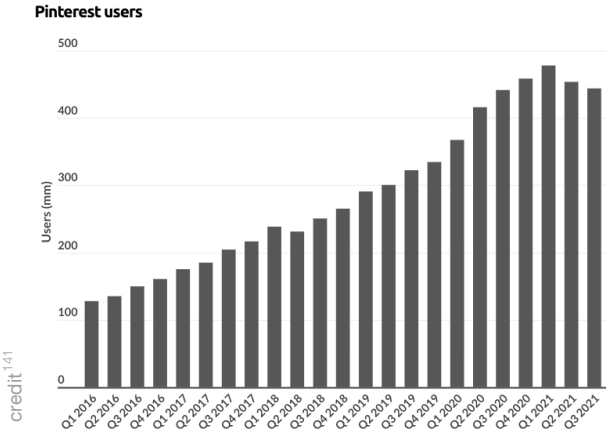


Figure 25: Pinterest is a type 1 “Viral” product that follows the logistic model

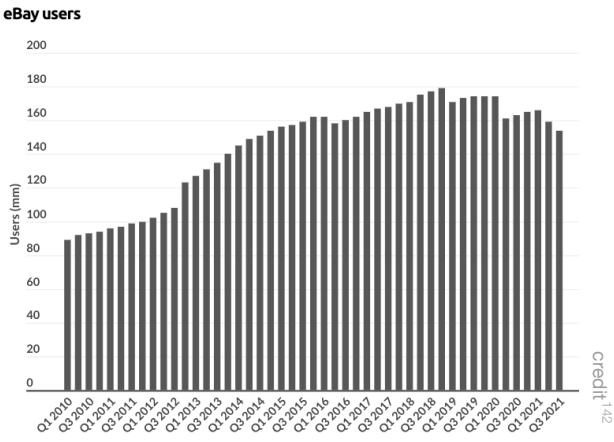


Figure 26: eBay is a type 2 "Word-of-mouth" in the number of buyers, following the logistic model (though also sagging towards the end, reminiscent of the Elephant Curve)

long-enough timeline, a decline. An Elephant with a more stretched-out trunk.

The Facebook Messenger product appears to exhibit at least the first half of the logistic curve (Figure 31).

Furthermore, this curve is actually a sum of US growth and outside-US growth. Looking only inside the US, Facebook Messenger is further along the curve, past the linear midsection and already leveling out near some carrying-capacity (Figure 32).

The same thing happens with Facebook DAUs and MAUs.* DAUs in the United States and Canada are logistical and have already topped-out at an apparent natural carrying-capacity of 185 million (Figure 33).

Breaking out MAUs by all geographies reveals that top-line growth of users is an aggregate of some geographies essentially not growing at

* Daily Active Users, Monthly Active Users

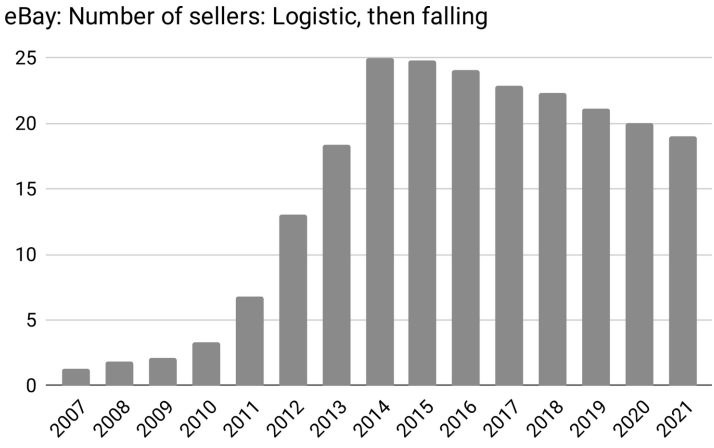


Figure 27: eBay also follows the logistic model in the number of sellers (with even more pronounced sagging)

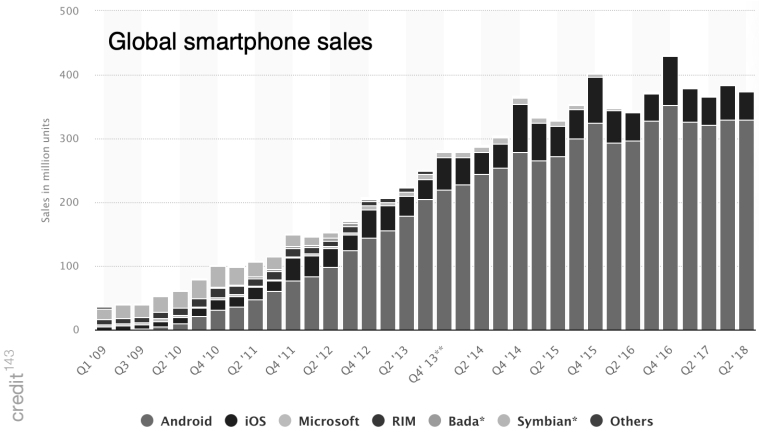


Figure 28: Smartphones are a type 3 “Hot Trend” that follows the logistic model all the way to saturation

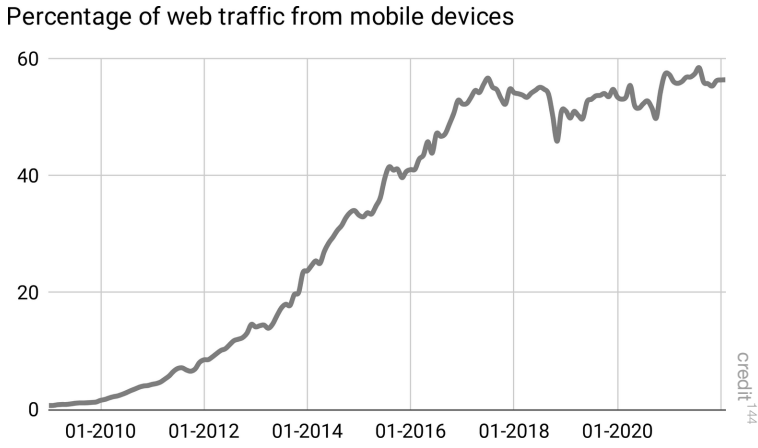


Figure 29: Smartphone *usage*, separate from smartphone *sales*, is also logistic. Many pundits predicted this percentage would grow nearly without bound; in fact it saturated at 55%.

all (late in the curve), while others are still growing, albeit also linearly (middle of the curve) (Figure 34).

The result of these individual effects of different products, released at different times, in different geographies, each with a "marketing campaign" style growth curve, is that it adds up to linear growth (Figure 35).

Does this conform to the Elephant Curve? Is this really still essentially quadratic? The answer is clear when we plot the same data, this time measuring the year-over-year change in MAUs. Not as a *percentage*, but as numbers (Figure 36).

Why do we keep seeing this pattern, even at the scale of Facebook, one of the most "viral" products of all time? Because mathematically, things that look like an Elephant Curve, even if the logistic "trunk" is elongated over time, are linear for nearly their entire lifetimes. Everywhere except the very beginning. Adding up linear things definitionally creates a quadratic.

As a striking example of this claim—that multiple, various Elephant Curves result in quadratic growth in the real world—consider

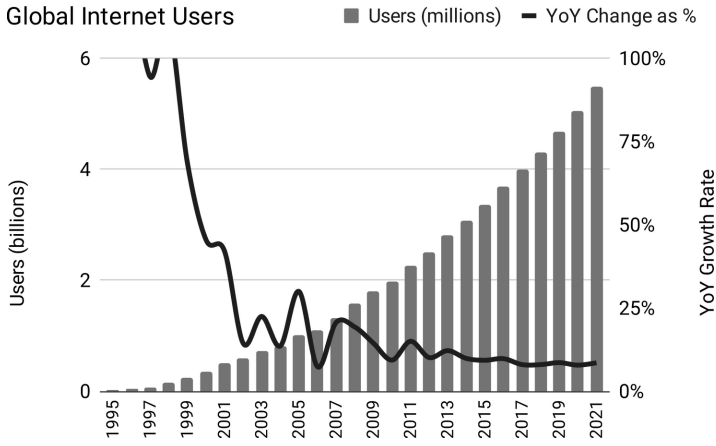


Figure 30: The internet is a type 3 "Hot Trend" product with a near-exact logistic shape; at 66% global penetration, it hasn't reached carrying-capacity, but it's been in its linear mode for many years, and fell off the exponential path sooner than you might have expected

the detail behind the earlier chart of Global Internet Users over time, a type 3 product. Every country has grown logistically, at a variety of starting-times, diffusion rates, and carrying capacities, yet the aggregate is quadratic (Figure 37).

To be certain the graph at the bottom (which is the same data as the chart shown earlier) is specifically quadratic, we chart the absolute difference in online population year by year. In a quadratic, these differences should grow linearly, i.e. each year adding a constant amount more than the previous year added. Which is indeed what we find, as precisely as we could expect from data in the messy real-world (Figure 38).

Bringing it back down to the scale of a single company, consider Netflix, another type 3 product. While their overall growth accelerates, under the hood we can see the US was already in a phase of slow-growth by 2014, with outside-US is taking up the slack through 2019 (Figure 39).

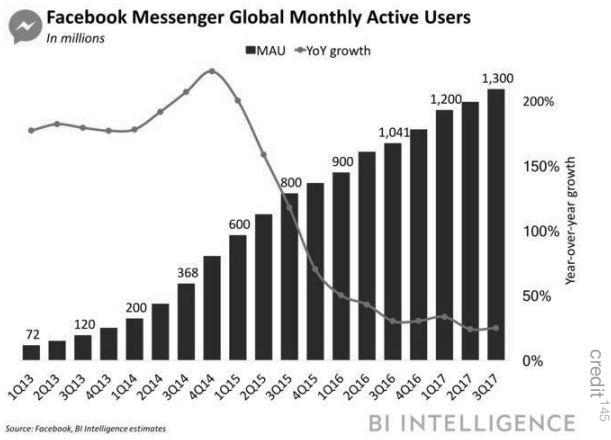


Figure 31

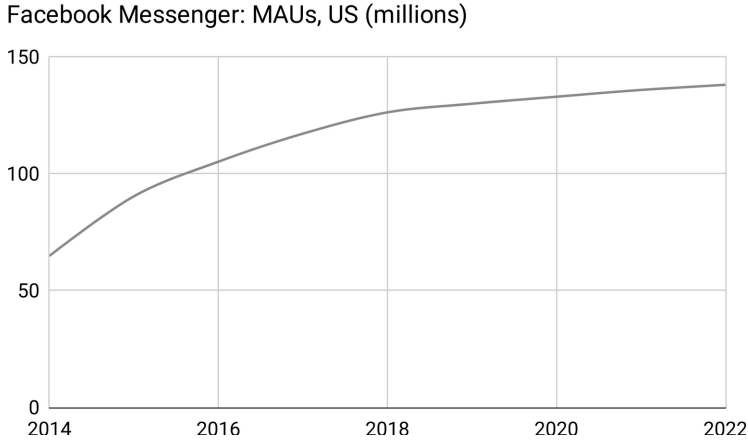


Figure 32

If we chart the *changes* in subscribers, rather than totals, it's even more clear that growth in the US has been in the declining phase of

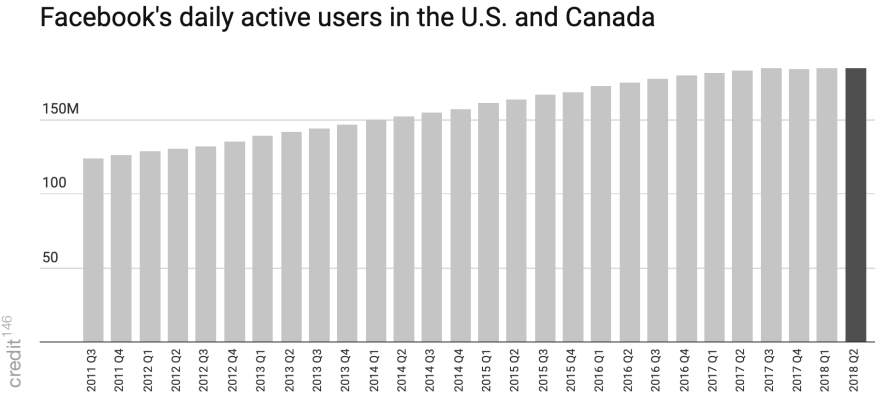


Figure 33

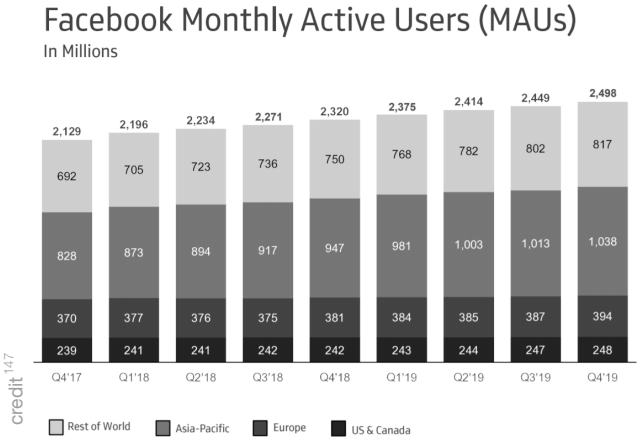


Figure 34

the Elephant Curve for a while, with outside-US is growing linearly (Figure 40).

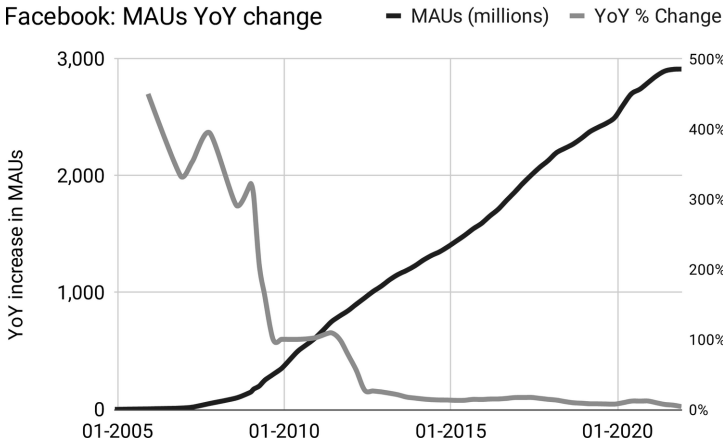


Figure 35: Exponential growth for the first few years crashes down into linear growth for nearly twenty years, from large-scale logistic-shaped products and geographies

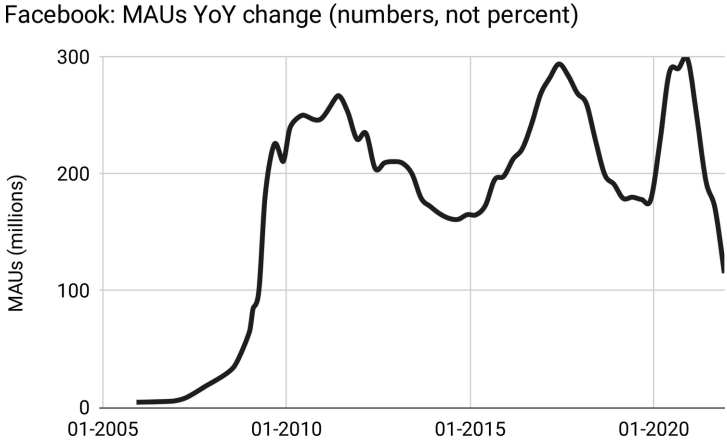
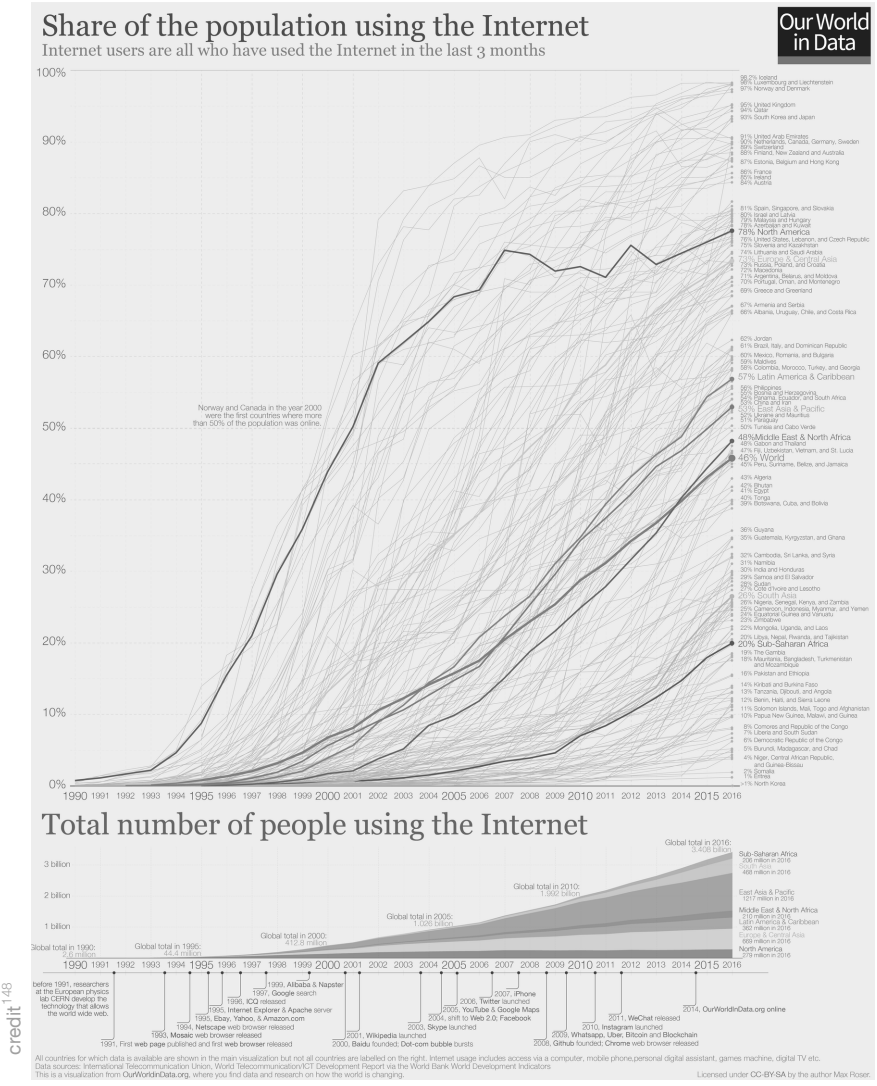


Figure 36: Facebook MAU growth is indeed an Elephant Curve: Logistic at first, then flat(ish), then starting to decline.



credit 148

Figure 37: Thirty years of varied logistic growth adds up to quadratic growth

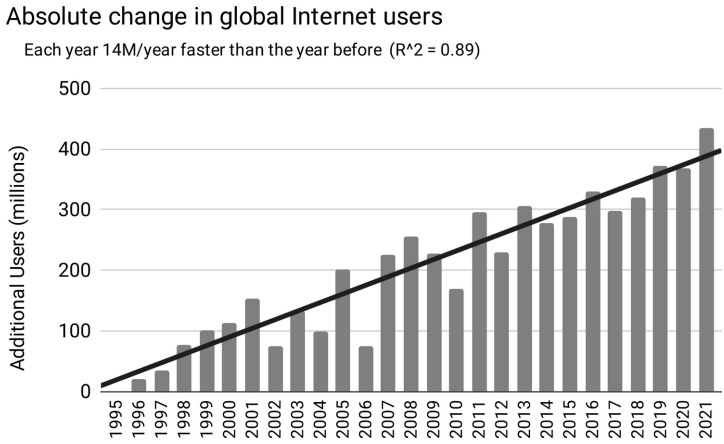


Figure 38

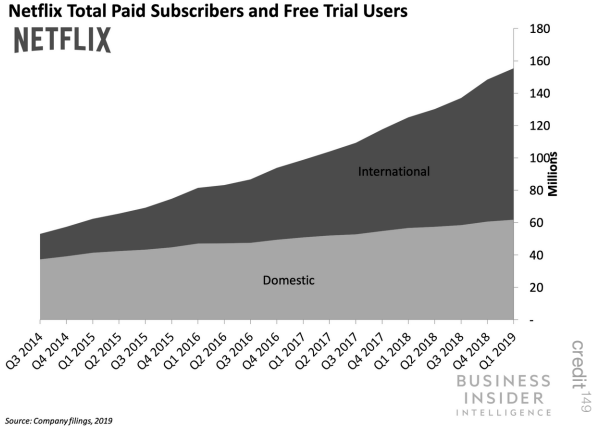


Figure 39: A quadratic top-line, created by two roughly-linear geographies

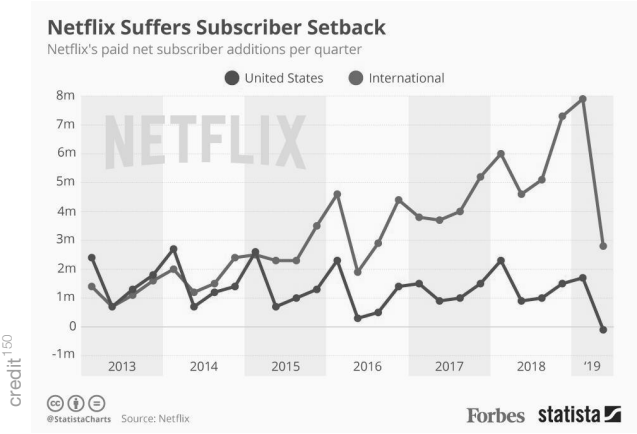


Figure 40

And then, fast-forwarding to 2023, looking at total, global subscribers, we see that growth slowed outside of the US as well, and the familiar Elephant Curve returns in its entirety (Figure 41).

Logistic growth with a varying carrying capacity: Start with market-share

Suppose you're Facebook, and you've saturated many markets. You might be at carrying-capacity for those markets, but more people are still coming online. The markets are growing, so your carrying-capacity is growing, so you should still be able to grow too.

Indeed, recalling the charts above, Facebook's current MAU growth rate, and that of global Internet users, both are currently hovering around 7% per year. Which isn't a coincidence.

Let's plot Facebook's MAUs as a percentage of people online—their *market share* (Figure 42).

Finally we have the complete answer to why Facebook's growth appears so "linear," when the theory expects an Elephant Curve. When

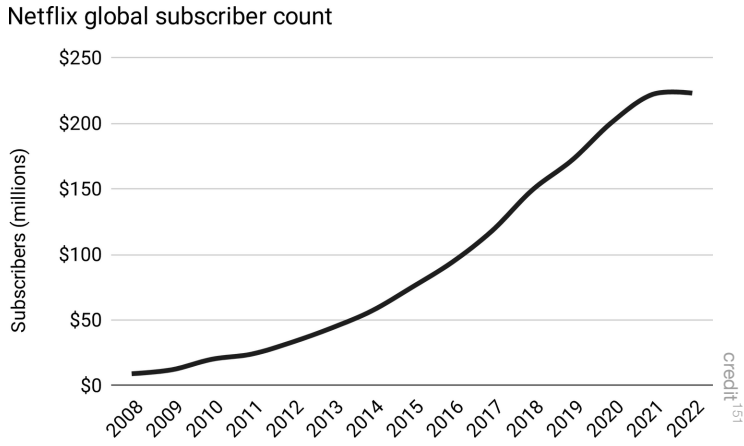


Figure 41

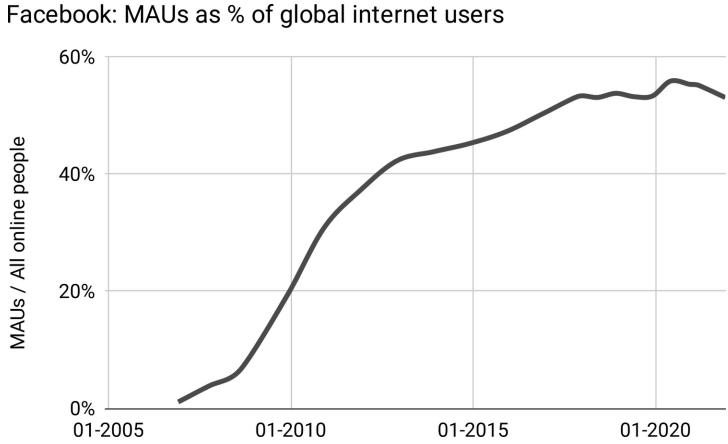


Figure 42: The Elephant Curve strikes again

you examine growth *relative to market size* it *is* an Elephant, complete with logistic trunk, optimized back, and declining rump (even despite a COVID bump).

This is why at-scale companies are willing to spend billions of dollars increasing the size of the market—it's one of the few ways to create growth other than raising prices. So Google spent billions on Loon—a subsidized service to bring low-cost internet to remote areas of the world. Its problem-statement is the first text on its website:[152] "Billions of people across the globe still don't have reliable, affordable access to the internet." Or, putting it another way, "Wifi balloons are a kooky idea but how else are we going to increase the carrying-capacity of the 'global internet user' Elephant Curve?"

Or Facebook with its "Free Basics" system that (in their words[153]) "Helps people discover the relevance and benefits of connectivity with free access to basic online services." Except actually it's only a few, hand-curated websites, all of which just happen to be western consumer products companies that are large Facebook advertisers, and the only available social network just happens to be Facebook. And there's no email, so I hope you like Facebook Messenger. In other words, a digital colonialism[154] whose purpose is to increase the carrying capacity of Facebook MAUs and the advertising that goes with it.

Elephant Curves are more visible when we plot growth as market share, because this incorporates the idea that carrying-capacity of the underlying market can itself be a moving target.

Logistic growth with a varying unit revenue

We've largely been analyzing *users* rather than *revenue*, and for good reason: The lifeblood of any product is people who use it, regardless how much money it can extract in the process.

However, when we turn to revenue, we find that curves can become perkier. Facebook's user growth might be linear, but could it be that revenue is exponential? It's certainly not linear (Figure 43).

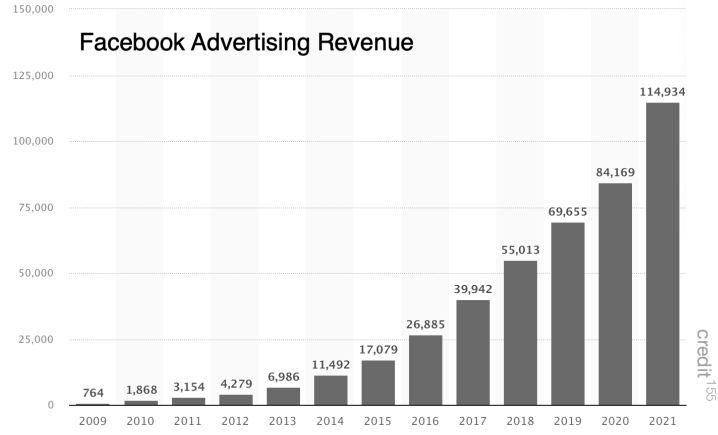


Figure 43

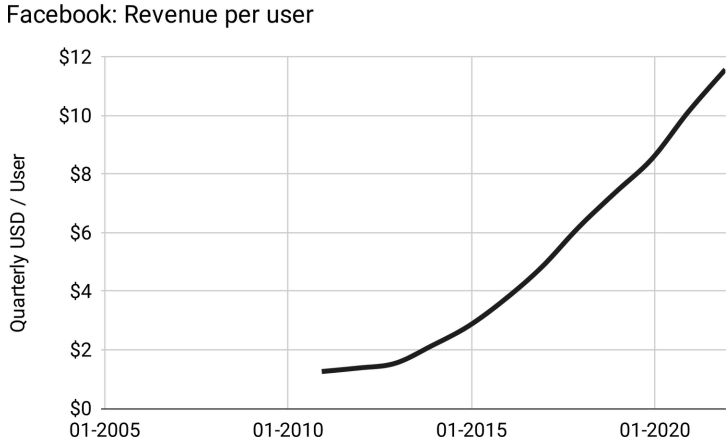


Figure 44: Logistic or Elephant yet again

We already know Facebook's user growth is linear, so the missing piece is Facebook's revenue *per* user (Figure 44).

Perhaps by now we're not shocked to see the Elephant Curve once again. And we also know how the rest of the story goes: Because MAUs are Elephantine (which means mostly linear), and revenue-per-user is Elephantine (which means mostly linear), when you multiply them you get a quadratic, not an exponential, and that's what we see in Facebook's overall revenue growth.

ACTIONABLE CONCLUSIONS

When we seek out the Elephant Curve in our marketing channels, product lines, geographies, and verticals, not just in its hopefully-explosive initial phase, but its phases of optimization and decline, we can proactively look for these phases, and take action.

Model by component

Our final discussion on the value of analyzing components of growth separate leads to a prescription for analyzing growth.

1. Estimate the growth curve for the entire market. Expect to be Elephantine (or simply logistic, in the case of trends that you can reasonably assume will not decline in the forecasted future, like global Internet use or smartphone use).
2. Estimate the product market-share curve. Expect to be Elephantine, and don't be so bold as to assume your product will never decline relative to the market—are you better at execution than Facebook?
3. Estimate monetization, i.e. revenue per customer.* This curve *might* be Elephantine, but not necessarily. It is highly dependent on the product and market, on how distinct the product is competitively, on the budgets of the customers, and more. Facebook has a strong moat (p. 761) (network effect) and doesn't charge end-users, so they (like Google) can raise prices consistently. A product in a commoditized market might never be able to raise prices, and thus must find growth in avenues like increasing usage, the introduction of companion products, expanding to other verticals or geographies, or by applying their technology to new markets.

You get better models by predicting each of these components separately, then multiplying for a final growth prediction. You're also better able to track the model against reality, as more data becomes available.

Besides this break-down, there are many operational ideas suggested by the results above, especially for managing marketing campaigns.

---

* The definition of "customer" should match whatever activity is most highly correlated with growth; this is also what "market share" should mean. For normal products people pay for, this is simply "paying customers," but for example in the case of Facebook, this is MAUs at least, perhaps even DAUs.

This might be expanded in a future article, but for now, these probing questions might lead to better ideas on how to analyze and affect growth:

Advice for Marketing teams

- Is our AdWords campaign topped out? Are we fooling ourselves into thinking there's more inventory to access? How much more optimization is there to be had, and how would know? Are we hitting a decline due to uneconomical auctions, and if so, what is our reaction? When should we start experimenting with new channels, rather than continue to flog the AdWords channel for results that don't exist?
- Is it OK to be less cost-effective if it means we can stave off decline? Should we be that "irrational bidder" who bids "too much" because we're wise enough to see value beyond immediate cash pay-back? If so, how do we quantify that value, so we know just how "irrational" to be?
- To hit our growth goals for the year, what would have to be true of the growth of existing campaigns? Which can be reasonably expected to grow, hold steady, or shrink, based on their phase? How many additional, successful campaigns do we need, and how soon? Since not all that we attempt will succeed, how many do we need to start to yield the final quantity we need?
- Should we lean into newer channels before others figure them out, saturating the channel and cause clicks to be both expensive and more rare?[156]
- Rather than stack up small, limited campaigns, is there something more substantial that could generate more total growth? A single, large new geography instead of many smaller ones? A single, substantial new sales mechanism (e.g. reselling) rather than more advertising? A different pricing model instead of an additional

sales model? Even if it takes 10x the effort, and possible even if it takes 10x the time, it might have 10x the results.
- Or the reverse—do we pull funds when we smell decline, rather than spending our time and money fighting a losing battle, accepting a short-term hit on top-line growth in exchange for more efficient growth? Do we try to stack up many smaller, more efficient campaigns, generating growth as a bulk effort? Each effort affects the top line only marginally, but conversely our growth is less sensitive to the decline of any one campaign.

Advice for Product Managers

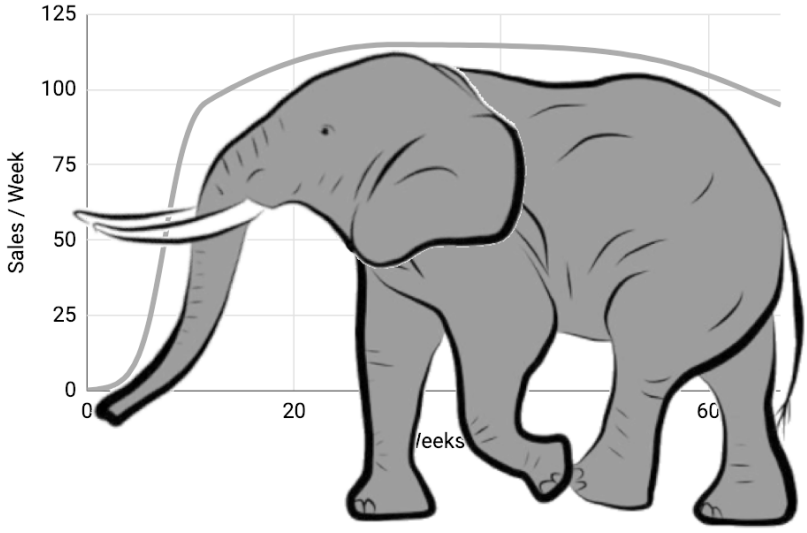
- It's great to add a feature to an existing product, but significant additional growth comes from increasing carrying capacity or creating a new avenue of growth. Early on you should focus on winning market share in one space, creating the first Elephant Curve, but after the product matures, something more drastic is required: Wholly new products, or updates significant enough to address new markets.
- It's well-known that companies need to add additional products to continue fast growth after achieving scale. However the second product is highly unlikely to achieve same market share and monetary scale as the first, so there needs to be multiple, not just one.* This requires serious investment, parallel efforts, and the chutzpah to kill off the ideas that aren't working.
- Because word-of-mouth-driven growth is so much more effective than marketing-driven growth (both in cost-per-customer and in that unlike direct advertising it grows automatically as the company grows), it is worth a great deal of time trying to figure out how to build that into the product, rather than relying only on the marketing team.

> *"Half my advertising is wasted. I just don't know which half."*
>
> *–John Wanamaker*

Mr. Wanamaker made his famous complaint more than a hundred years ago; even with modern analytics, today it's worse.[157] The quadratic growth model won't solve that puzzle, but the better you understand the mechanisms of growth, the more it is under your control.

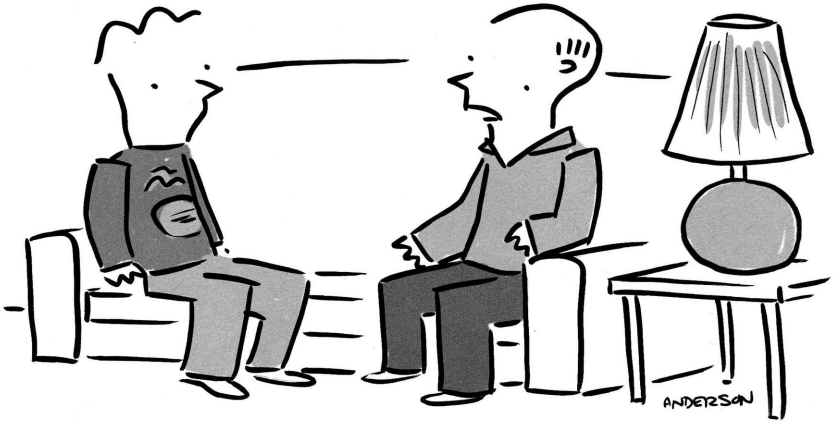
* This is true at any scale—advertising is still 82% of Google's revenue; of that 71% is advertising from search alone (i.e. excluding YouTube and other properties). Apple revenue is 60% iPhone. Even at smaller scales: Basecamp (neé 37signals) built multiple products over nearly two decades but only their first was successful enough to be worth working on; the company divested itself of the rest and rebranded to be identical to that product. It *is* possible for second products to eclipse the first; the iPhone was of course not Apple's first product; The Tesla model 3 outsells the earlier model X, And at my own company Smart Bear our second product ended up being 95% of sales, and we essentially did the same as 37signals and went to a single product model.

The Elephant Curve



Chapter 7:
Stubborn Visionaries & Pigheaded Fools

PUZZLE · SOLUTION



"Son, there comes a time in each man's life when he needs to decide which news he's going to believe."

THE PUZZLE

Scenario 1 (S1)
At time (A) you start an AdWords campaign.
At time (B) it's obviously not working; a waste of time and money.
...But you keep trying, and by time (C), it's working! **You did it!**

Scenario 2 (S2)
At time (A) you start an AdWords campaign.
At time (B) it's obviously not working; a waste of time and money.
...But you keep trying, and by time (C), it's still not working, and you've wasted even more time and money. **What a waste!**

We've all experienced both scenarios, not just in AdWords but in life in general.

But we misunderstand it.

S1 we call "**success through perseverance**," and you've heard this echoed in many platitudes. Winners never quit,[159] and quitters never win. Failure is a step (p. 1261) on the path to success. Failure is a pivot away from success (p. 193). Learn from your mistakes and next time you will succeed. Fake it 'till you make it. The thing all failed startups have in common (p. 379) is that the founders stopped trying.

S2 we call "**failure through obstinance**," and you've heard this echoed in many platitudes. Doing the same thing expecting different results is the definition of insanity. Those who ignore history are doomed to repeat it. Those who cannot be introspective (p. 845) and honestly see things as they are (p. 657), will fail. The thing all failed startups have in common is that the founders didn't pay attention[160] to what was happening outside their own egotistical worldview.

Are these the correct conclusions? No, they are convenient rationalizations.

Consider S1 and S2 at time (B). Up to this point they are identical. So how do you know, *at time (B)*, which scenario you're in? Because if you're in S1 you'd be a fool to stop, but if you're in S2 you'd be a fool to continue. How do you know whether you'll end up as a cautionary tale of someone who couldn't let go when they were clearly wrong, or as a hero who bravely fought through doubt to prove everyone wrong?

Maybe you shouldn't find out! Just stop at (B). No again, because if you're on the path of S1 you've lost your win. If you were on S2, you were "smart" to stop, but *either way you've failed* to achieve something useful. Stopping is *sure failure* while persisting is at least *possible success*. Stopping means you'll never create something great.

So you cannot know. Not for AdWords, not for product design (p. 853), not for the vision of your company and the market you hope to create around it (p. 71), not for almost anything, big or small. It all looks the same at point B (p. 429).

Venture capitalists don't know either, though it's their job to know. They're smart and do this for a living, but it usually doesn't work; most VC portfolios lose money.[161] Not even the experts know which path you're on.

The typical, backward-looking interpretation of these two scenarios is not the best way for us to understand the choices in front of us today, nor to evaluate our decisions in hindsight (p. 1253). It's not even clear that we've "learned anything," whether the outcome was good or bad.

Perhaps all we've done is made some choices and observed some results, and that's the end of it.

You could read this as depressing, because nothing is predictable and even the wisdom we believed we accumulated along the way is false wisdom. But clearly this point of view isn't exactly true.

So, read this as a positive, and realize that it liberates you to make decisions more easily—with less second-guessing in the moment, and less guilt afterwards. (And taking less credit when things go well (p. 449).)

When you realize you cannot know which scenario you're in, you realize that the job is to **find out which one it is as quickly as possible**, which means to cease your dithering, make a strong decision, keep your eyes open, try to measure what's happening (p. 645) as objectively as possible, hope for S1, but allow for S2, to not feel guilty if you guessed wrong, and not feel cocky if you guessed right.

Still, there are some guiding questions that can help you suss out which path you're on, and thus what to do.

SOLVING THE PUZZLE

These are my specific strategies that thrive under uncertain conditions (p. 193). Because many things in life are uncertain, these are often useful.

Besides those strategies, the following tactical questions help you determine which path you are on. Readers have contributed wisdom that I've lightly edited:

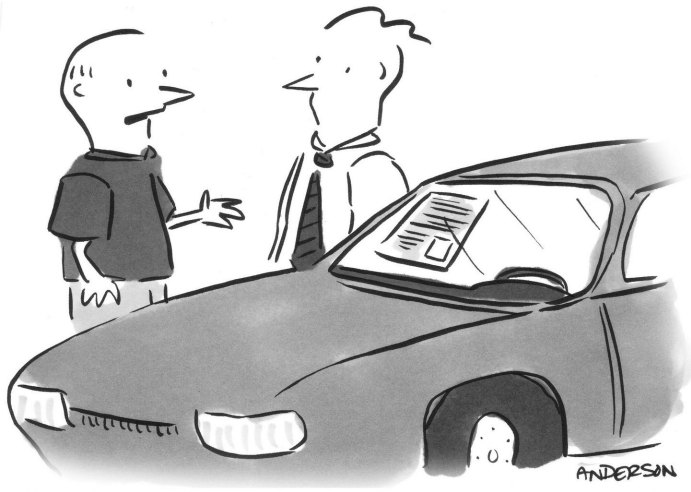
- **Timebox.** Set a hard deadline for how much longer you'll work on it without seeing improvement. (@farezv[162])
- Are you still **enjoying** the project and **learning** something from it? (@colemank83)[163]
- **Rate of progress**: Is the rate of progress slowing or accelerating? (@Daanlo[164])
- Use an **external sounding board** to keep you honest. Often the numbers won't tell you to go left or right. And even if the numbers show clear signs, we often ignore them because "they are outliers" and "this time is different." (@igriff[165]) (Ton Dobbe[166])
- Don't allow **sunk cost** to decide. (@igriff[167])

- If the team has **run out of ideas and conviction**, consider pivoting. If the team has run out of ideas and conviction for pivoting, it's time for a full reset or quit. Do you **see yourself doing this in 12 months?** If not, you might as well stop now. (@sachinrekhi[168]) (@awoodsnet[169])
- Is the product **noticeably better** than alternatives in the market or will be there soon? If the answer is no to both, consider stopping. (@yassin_baum[170])
- **Existence proof**: If others somehow achieved a similar outcome, then you shouldn't stop. Especially if their solutions are bad and yet they're making money. (@mibenz95)[171] (@nickresreal[172])
- Is it (a) **worthwhile**, (b) **fixable**, and (c) you **still have energy** for it? (@nurijanian[173]) Am I truly enjoying what I'm doing? Do I have room for improvement? Am I willing to put in the work to improve? Am I marketing this venture/skillset to the best of my ability Would my time be spent better trying a new project? (Alex Finn[174])
- **Opportunity Cost**: Are there opportunities you're missing because you're fixated on this? (@Liscoomi[175])
- **Consequences**: What's the downside if you don't finish? What do you get if you're successful? Are there smaller wins you can achieve along the way? (@awoodsnet[176])
- **Don't decide out of fear**. If you're scared of the outcome a little bit, stick with it. (@pascallaliberte[177])
- **Penny in the air**: Go for a long walk and listen deep inside. Often I already know deep inside, but just don't want to admit it to myself (p. 657). (@awoodsnet[178])
- **Go back to "The Why"** that set you off on your journey to begin with. If you now have more information to assess the credibility of that why, re-assess. If you still have the same information available and still believe in the why, press on. (@temlabs[179])

In any case, just shut up and get going.

And when you're done with that, don't look back too much, just shut up and get going again.

Chapter 8:
How repositioning a product allows you to 8x its price



"I like it, but I'm looking for more of a status symbol. Any way you can double the price?"

Pricing is much more about positioning and perceived value than it is about cost-analysis and unconvincing ROI calculators.[181] Let's see how repositioning can result in a much higher price for the same product.

You've created a marketing tool called DoubleDown that *doubles* the cost-efficiency of AdWords campaigns. You heard that right folks—as a marketer, you can generate the same impact, the same number of conversions, the same quality of sales leads, but with *half* your current ad-spend. Wonderful! Who doesn't want higher ROI.

What can you charge for this tool? Well, the customer will save a certain amount of money on ad-spend; surely you cannot charge more than that. Let's say you can charge 25% of the savings and still find many willing customers (p. 71).*

Here's what your sales pitch looks like to a customer who spends $40,000 per month on AdWords (Figure 1).

Great deal! The VP of Demand Gen will be able to boast to the CMO that she saved the company $15,000/mo even after paying for DoubleDown, and you're raking in a cool $5,000/mo. Everyone's happy!

Now let's see **how to charge eight times as much money for the same product**.

Marketers have a paramount goal: Growth. Even indirect marketing like brand, events, and PR have the indirect goal of supporting growth. In the case of DoubleDown's customers it's direct: Growth through lead-generation through AdWords.

Increasing growth is much more valuable than decreasing cost. To see why, consider the following two scenarios:

* There is data supporting the "25%" number. In addition, there are these observations: If you charge less, you're not monetizing enough; if you charge much more, you leave too much space for a competitor.

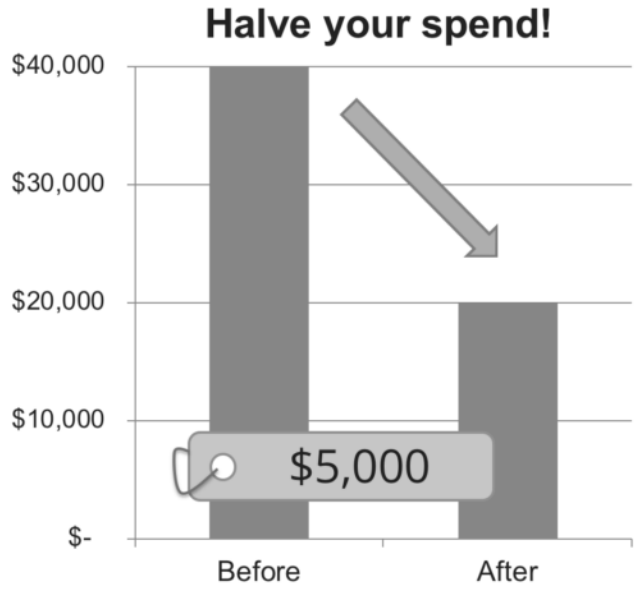


Figure 1

1. CMO reports to the CEO: **I was able to reduce costs 20% this year.**
   The CEO is happy. The CEO's follow-up question is: How will we use those savings to grow faster?

2. CMO reports to the CEO: **I was able to increase growth by 20% this year, but it also cost us 20% more to achieve.**
   The CEO pumps her fists, releasing peels of joyous laughter. The value of the company increases non-linearly. The additional revenue growth more than pays for the additional marketing cost that generated it. The CEO's follow-up question is: Can we spend even more? How can we ensure this happens again next year?

It's always 10x more valuable for a business to *grow faster* than it is for the business to *save money*.

This insight points us to an alternate pitch for DoubleDown. It's not about spending less for the same amount of growth, it's about spending more to create more growth.

Continuing our example, suppose the customer generates 200 quality sales leads per month from their $40,000/mo spend. The sales pitch changes as follows:

You're paying $200/lead right now, yielding 200 leads per month. Using DoubleDown, you can *double the number of leads* you're generating, still at a cost of $200/lead (Figure 2).

The customer is willing to spend $40,000 to generate 200 leads, and therefore is happy to spend $80,000 to generate 400 leads. It doesn't matter how much of that $80,000 is going to AdWords versus going to DoubleDown. The goal is not to "save money on AdWords," but rather to "generate more growth at a similar unit cost."

Figure 2

In the "saves money" pitch, the value was $20,000, and the customer needed to keep 75% of that value-creation. Whereas in the "generate growth" pitch, the value is $40,000, and the customer is happy to pay 100% of that value-creation to a vendor. **Both the amount of value created, and the percentage of value the customer is willing to pay, is a *multiple* higher for the "growth" pitch versus the "save money" pitch.**

So the next time you want to formulate your product as a way to "save time" or "save money" or "be more efficient" .... don't.

Instead, figure out how your product creates value in the way your customer already measures value, and position your product as a way to accomplish *that*.

Chapter 9:
Fermi ROI: Fixing the ROI rubric

RUBRIC FAILURE · THE FERMI SOLUTION
PUTTING IT TOGETHER · BREAKING TIES

"You've been traded for some big data,
two spreadsheets, and an algorithm."

credit 182

"Delivering maximum value in minimum time" sounds wise, but it's not exactly a stunning insight. It's not like we're *trying* to take an eternity to deliver trivialities.

It is more analytical—but no more insightful—to insist on "Maximizing ROI." Return-On-Investment is some measure of "value," divided by some measure of "time," yielding some measure of "productivity."

With good intentions and this reasonable-sounding goal, we reach for the rubric: A spreadsheet of inputs of numeric "value" and "time," a calculation of the metric of productivity, and an output where the best ideas are sorted to the top.

INVISIBLE FAILURES OF THE RUBRIC

Unfortunately, the rubric fails us for several reasons. The failures are not obvious, so we don't realize they're happening. Explaining the traps will lead to a framework that avoids the traps.

"ROI" contains more noise than signal

Consider a feature that ended up producing 20% less impact and taking 50% more time than originally expected—a common outcome in the real world:

	Impact	**Effort**	**ROI**
Estimated	60	4	15
Actual	48	6	8

In this reasonable scenario, it ended up producing *half* the ROI than originally calculated. Therefore, in retrospect, the initial ROI estimate was largely noise. Apply this observation to a typical ROI rubric:

	Impact	**Effort**	**ROI**
Feature A	34	2	17
Feature B	60	4	15
Feature C	36	3	12
Feature D	10	1	10

The spreadsheet suggests (A) is the winner, and in particular that (A) is clearly better than (D). But neither of those statements are true. If (A) could easily result in half its stated ROI, there's no objective justification to claim its superiority over (D), or over any other item.

Shocked at this revelation, we could react by ignoring the lesson. We could justify (A)'s supremacy by telling ourselves "estimation errors cancel themselves out." But the errors do not cancel out. Effort is almost always under-estimated; 4-sprint projects that stretch into 6 sprints are not "cancelled out" by an equal number of 4-sprint projects that are completely finished in 2 sprints.

"Impact" is ill-defined

No matter how you measure impact, your numbers end up far less precise than they appear in the spreadsheet.

It's difficult to predict (p. 193) numbers like "revenue-increase due to feature X." Teams use various techniques to address the uncertainty, but in every case the number in the spreadsheet contains significant error:

- *Revenue we would get from existing customers currently asking for this feature.*
  That's precise but often uncorrelated with how much money we *will* make from it in the long run, so it's not an accurate measure of impact.
- *Additional sales we would make in the next year if we had this feature.*
  This number is low-confidence, high-error. Even in hindsight it's hard to attribute revenue from a single customer to a single feature; if it's unmeasurable *post hoc*, surely it's unpredictable *ex ante*.
- *Use a relative scale, e.g. "1-5," calibrated by previously-created features.*
  Defined as "1 is like feature X, 5 is like feature Y." Not only is this a wild guess, different people will interpret the scale differently, yet answers differing by a single point generates a large variance in computed ROI.

It's even worse when want to combine multiple metrics of impact (p. 645), e.g. revenue, unit-profitability, and retention. Each contains intrinsic error, then a mathematical combination compounds the error, then the ROI calculation expands the error yet again.

You can't tell from glancing at the spreadsheet, because input numbers appear typical and output numbers boast many digits after the decimal place. But those digits aren't indicative of precision. They're a random number generator.

THE FERMI SOLUTION

These traps dictate requirements for a solution. We need scores containing dramatically more signal than noise. The scores must be

well-defined, which means their definition and estimation should be obvious and uncontroversial.

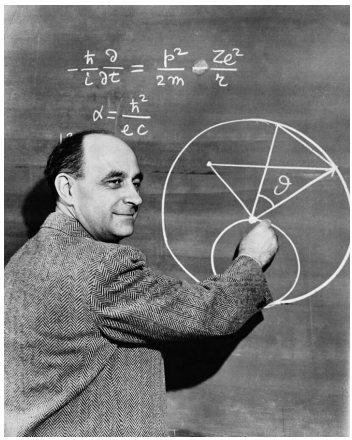
Only Fermi estimations allowed

The first full-scale nuclear bomb was detonated at 5:29am, July 16, 1945, in the New Mexican desert of the United States. The physicists who invented it were huddled in a truck behind a plate of welder's glass to reduce the radiation to non-lethal levels.

The physicists were already causing trouble. Future Nobel Prize-winner Richard Feynman inexplicably decided to observe the blast without eye protection, causing frightening but ultimately temporary blindness. Current Nobel Prize-winner Enrico Fermi had taken bets with military guards about how much of the atmosphere would ignite, and whether it would incinerate the entire state or the entire world; some of the guards asked to be excused from the base, angering the project director.

Fermi was also interested in the amount of energy released by the blast—one of the main goals of the test. Not wanting to wait for official analysis, he made his own estimate on the spot, using a technique that now bears his name, and that we will use to fix our rubric:

> About 40 seconds after the explosion the air blast reached me. I tried to estimate its strength by dropping from about six feet small pieces of paper before, during, and after the passage of the blast wave. Since, at the time, there was no wind I could observe very distinctly and actually measure the displacement of the pieces of paper that were in the process of falling while the blast was passing. The shift was about 2 1/2 meters, which, at the time, I estimated to correspond to the blast that would be produced by 10,000 tons of TNT.
>
> –Enrico Fermi, Top Secret interview[186] July 16, 1945, declassified in 1965



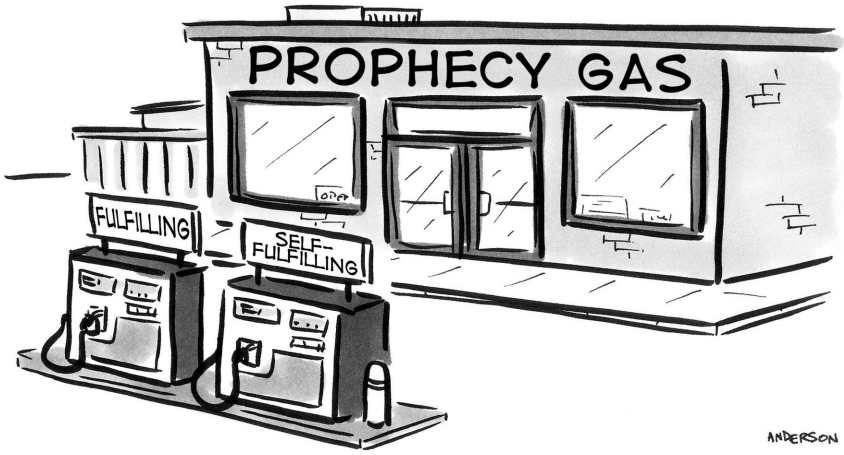
"Enrico Fermi[183]" by Argonne National Laboratory[184] is licensed under CC BY-NC-SA 2.0.[185]

The official estimate of the energy output of the blast was 21,000 tons of TNT. Fermi's estimate was surprisingly accurate given such inaccurate input data and quick, simple, mental calculations. How did he do it?

The trick—useful everywhere in life—is to estimate values using only orders-of-magnitude, a.k.a. powers-of-ten. No "low/high ranges," no precision, not even any digits other than a 1 followed by a quantity of 0s. It sounds far too imprecise to be practical, and yet Fermi's bits of paper demonstrate that it just might work.

Joel Spolsky famously[188] loved an interview question built for Fermi estimation: How many gas stations are there in Los Angeles County? "I don't know," although accurate, fails the interview. Fermi estimation, succeeds.

- There are 10 million people in the LA area *(or at least, more than 1 million and less than 100 million, so by the rule of "only powers-of-*



credit[187]

ten," our number is 10 million. The actual number in LA County happens to be 10 million)[189].

- 1 car per person *(or at least, not 10 and not 0.1. The actual number is 0.54)*[190].
- A person refills 1 time per month *(the actual number is closer to 2.5/month, but it's not 10 and not 0.1).*
- A gas station handles 10,000 refills per month *(200 per day yields 6000 per month; it's not 1000 nor 100,000).*
- With 10 million refills per month (10 million cars with 1 refill per month), divided by 10,000 refills per station, 1000 gas stations are required.

Despite being wildly inaccurate in detail, the end result of 1000 gas stations is indeed the nearest order-of-magnitude to the actual number of 600.[191] Surprisingly accurate, considering we arrived at an answer in a minute or two, without looking up a single number.

What would happen if we used only Fermi estimates in our ROI calculator?

An immediate result is that most numbers are trivial to estimate. Note how easy it was to decide the values in the gas station example,

because the two adjacent power-of-ten choices were definitely wrong. It's nice to be so confident! Relish that.

Another result is that we've satisfied one requirement generated from the failures of the rubric: Scores are either identical or *very* different, causing calculation error to be small in comparison. The values are largely obvious and uncontroversial, if only because our artificial constraint on their values makes the adjacent choices absurd. This is progress.

Sometimes the numbers are still controversial, e.g. *Could this product produce $1M in the next few years, or actually $10M because it would dramatically increase both retention and our ability to win new deals?* But this debate is a wonderful use of our time, because it means people have wildly different assumptions, or different levels of optimism, or a different understanding of the customer, the market, or the idea. So another constructive result of Fermi estimation is that we spend our debate-time on strategic-level discussions, and no time on areas where we disagree in small degrees that are in any case overwhelmed by typical estimation-error.

In exchange for this progress, we lose precision. And admittedly, sometimes we *do* possess precise input data. For example, maybe we could look up the fact that the average number of cars per person is `0.54`, and use that figure instead of `1`. Surely it's wise to retain precision whenever possible? Yes, but only if you recognize that, after doing math with other Fermi estimates, the end result is still not more accurate than the power-of-ten. That is, even using `0.54` instead of `1`, as soon as you combine it with your extremely inaccurate estimate of 10,000 refills-per-station, you still have to zero-out all the digits in the result except the first one. You must meticulously discard the noise; if you're diligent about that, your estimation is more accurate. Otherwise, maybe you should just enjoy the simplicity of the basic Fermi system.

So we've satisfied the first requirement. What about the final remaining requirement, which is to clarify the definition of "impact?"

Fermi Impact: Quantitive edition

Fermi estimation solves the puzzle of how to estimate the revenue impact of a sizable feature. We can pick any definition we want, but limit ourselves to Fermi estimation, and suddenly it becomes easy, or at least a simpler discussion. Any of these definitions could work:

- This feature will increase revenue by: $1k/mo, $10k/mo, $100k/mo
- This feature will increase new customers/month by: 1, 10, 100, 1000, 10,000
- The take-rate of this feature in our existing customer base would be: 1%, 10%, 50%
- The number of customers who would actively use this feature would be: 0.1%, 1%, 10%, 50%

We cheated a little in those last two, because "100%" is not a possibility. Still, a 5x separation is pretty good; the danger is when the difference is only 2x or 20%. This is still a much wider spread then, say, Fibonacci estimation[192]).

It's clear how this works with quantitative measures, but what about important things that are not numbers?

Fermi Impact: Qualitative edition

How you score things which aren't numbers like "brand development," "competitive advantage," or "customer delight?"

The following is a real example from a few years ago, when WP Engine was launching a new product line called Atlas.[193]* We decided to build a thought-leadership presentation that would engage software developers on topics relevant to the new product line. We wanted it

* Tagline: The complete Headless WordPress Platform for absurdly fast dynamic sites.

credit[194]

MIKE REALIZES HE'S ONLY BROUGHT HIS 'C' GAME

to be genuinely intriguing and useful. We brainstormed twenty-two topic ideas, all pretty good at first glance. How should we select just four or five topics, given that "intriguing" and "useful" are qualitative and subjective?

First, we decided on our goal, even if impossible to measure, or not even a number. The primary goal was for **everyone to come away loving the content and interested in our software**. We broke this down into four sub-goals, where the overall goal would be achieved only to the extent that *all four* of the sub-goals were achieved:

1. Topic is widely applicable
2. Topic matters to real people
3. Topic is insightful (i.e. non-obvious to a typical practitioner)
4. Topic is relevant to specific features or capabilities in Atlas (i.e. self-promotion)

The next step was to convert these concepts into something we could "score," i.e. specific questions that a person could contemplate and debate with another person. Otherwise, two people will have different ideas of what each goal means. The answers still won't be *numbers*, but at least the questions are specific:

1. How many customers care about this topic?
2. How *much* do *those* customers care?
3. How insightful is our perspective on this topic?
4. How powerfully or uniquely does Atlas pay off the insight?

Because the answers aren't things you can measure with numbers and units, we had to use some sort of "1-5" scoring system or, to take the Fermi lesson, powers-of-ten numbers. The key is to *be specific* as to what each number *means*, otherwise two people will have different interpretations for a phrase like "Insight is a 2 out of 5."

1. How many customers care about this topic?

Fermi Value	Definition
100	Definitionally everyone
70	Most
30	Some
0	I can think of one or two
0	No one

2. How much do those customers care?

Fermi Value	Definition
1000	Mission-critical to the success of their project
100	"I care; this is a serious concern we've discussed internally"
10	"I'm curious, tell me more"
1	"Meh, whatever"
0	"Don't waste my time with this"

3. How insightful is our perspective on this topic?

Fermi Value	Definition
1000	"Wow!!! This changes everything."
100	"I took notes, thanks, that's a great point."
10	"Yup, OK, makes sense."
1	"No shit, Sherlock."
0	"No clue what you're even talking about."

4. How powerfully or uniquely does Atlas pay off the insight?

Fermi Value	Definition
1000	"Whoa, I'll buy Atlas just for that alone!"
100	"Ohhh, nice, OK I'll take a look."
10	"That's fine."
1	"So what?"
0	"I actively do not want this."

The combination of Fermi values and specificity in both the questions (rather than broad concepts) and values (rather than "1-5") made it easy to score. Because "all four" scores were required to deliver on the original goal, multiplying the scores* sorts the best to the top.

It worked; the presentation was routinely scored "5 out of 5" in post-event surveys,** and the biggest complaint was a desire for even more detail. That is a "complaint" indicative of success: It means "I want to hear more from you," and therefore is further evidence that the choice of topics was spot-on.

Fermi time estimates

We've explored how to estimate "impact" with Fermi-approved values, but what about time-estimates? Those also exhibit large errors. They, too, need the Fermi treatment.

Nearly twenty years ago, before Scrum defeated rivals like Extreme Programming to become the *de facto* standard for waterfall-averse software developers, our engineering team at Smart Bear Software used a simplistic but effective method of estimating work. Our software was installed on-premises by customers (this was The Time Before SaaS), so "continuous deployment" was impossible. Instead, we made a few major releases per year, supplemented with minor bug-fix releases. We planned about four months of work at a time.

We scrawled each idea on an index card, with only enough space for a title and a few bullets about scope and intent. We placed time

---

* The mathematically-inclined reader might point out that because all values are powers-of-ten, the operation "multiply everything and sort" yields exactly the same result as "Number these 1-5, then add everything and sort," because the latter is just the logarithm of the former. Why, then, bother with Fermi values? (a) As in the first question, specific values (as opposed to strictly only powers-of-ten) might have a valid meaning, which creates different values from "1-5." This "breaks ties" in a meaningful way. (b) By emphasizing that the values are designed to be widely divergent, we generate better specific phrases, and more agreement in scoring each topic, and therefore hopefully a more accurate output.

** Oh the irony of scoring the presentation 1-5 after everything we just said!

credit[195]

"So, where do you see yourself in ten minutes?"

estimates in the bottom-right corner, but the only three choices were: `2d`, `2w`, `2m`—two days, two weeks, or two months.* Whatever you think the real estimate is, round up.

The overwhelming majority of cards generated no disagreement over the estimate; this saved us hours of analysis and debate. The controversial ones were always a matter of definition and scope, which is a typical and important conversation to have, regardless of your system for estimating work.

To select the items and plan team capacity, we spread the cards across a table, and each person held the set of cards they were going to execute. The project time frame was measured in weeks; a typical window was 16 weeks. We used a rough conversion that two `2d` cards equals one week, and one `2m` card equals eight weeks.** It was easy to

* I hear you yelling "No! Story Points!" Hang in there, it's OK, this was long ago in the Third Age when the Elves still abided in Middle-Earth. Elves don't use story-points.

** You might wonder why we didn't just use a mathematically identical scale like `0.5w`, `2w`, `8w` or some abstract unit of "effort" that could scale like `1`, `4`, `16`, or a Fibonacci number of weeks. The reason for using real time is that we felt it

see who was over- or under-scheduled, and then load-balance according to capacity and ability.

If a project required multiple cards (nowadays we'd call that an "epic" or "big rock"), we marked that set of cards with a color. The idea is that either *all* the cards of that color should be accepted, or *none*, otherwise we would have spent a lot of time on something without having shipped a usable feature.

Here's the punchline: It typically took a few hours to create the entire plan, and four months later we typically hit the plan within a week of the original estimate.

In retrospect, it's a form of Fermi Estimation. It's not *exactly* powers-of-ten, but measured in work-days it's close: `2` vs `10` vs `45`. Essentially powers-of-five, it still maintained the key idea of Fermi—to have so few choices that the correct one is easy to identify—while also being chunks of time that a human can relate to.

PUTTING IT TOGETHER

Let's combine these ideas, do the math, and decide whether it's helping us make better decisions.

Single-valued impact

Suppose we're picking features to implement, and the only metric of value is the revenue we believe that feature will generate in the next twelve months. We'll convert "effort" into work-days as given just

was easier to imagine than abstract units. The reason for keeping the quantity of values to three is the Fermi reason: This minimizes controversy and forces you to think critically about scope.

above, and round ROI to the nearest 10% to make it easy to read. Here's a typical result:

	NTM Revenue	Effort	ROI
Feature A	$100,000	2m	2200
Feature B	$10,000	2w	1000
Feature C	$1,000	2d	500
Feature D	$10,000	2m	200
Feature E	$1,000	2w	100

This has some nice properties:

Biased towards more impact
Products A and B are examples of "takes more time, but generates more impact." The rubric prefers more impact, even compared to "quick wins." This is a good bias, especially since impact in reality is often less than we had hoped.

Still correctly ordered if we misjudged
Consider what happens if we misjudge impact by 20% and effort by 50%—the original hypothetical from the beginning of this article. If Feature A delivered only $80,000 and took three months, the actual ROI would be about 1200, still ahead of Feature B. Or if Feature B delivered $8,000 in three weeks, actual ROI is 533, still ahead of Feature C. Therefore, typical misjudgments are not changing our decision. We fixed that deficiency.

Easy to explain to others
It's important to be able to explain your decisions crisply. The typical ROI rubric does not; the explanation is that one item slightly edged out another item, which isn't confidence-inspiring, and doesn't sound like a strategic decision. In contrast, Fermi ROI is easy to summarize; for example, why Feature A was selected: *Feature A has by far the greatest potential for impact. The impact is so large, it's worth spending multiple months on it.* Or why Feature B wasn't selected: *Feature B would be faster to implement, and therefore is a great idea, but Feature A's impact is so much*

larger, it's still the better choice. If we get part-way into Feature A and realize it's much harder than we thought, or much lower impact than we thought, then we'll pivot into Feature B.

Multi-valued impact

In the "presentation" example, we had four impact scores, which we multiplied together. The reason it made sense to combine them in that way, is that *all four are required* to achieve the goal (as explained previously). When you have a case where multiple factors are needed *together* to achieve a singular goal, then multiplying is the correct way to compute the score for that goal.

A common mistake is to use this formula when there are scores for things which are *not related*. For example, suppose you're wanting to maximize revenue, and also customer delight. You could make a Fermi score for each, but how do you combine them to get an ROI?

There are two answers, depending on your philosophy:

Do not combine unlike attributes
There is no way to compare "revenue" with "delight." These are different units of measurement, so any combination is nonsensical. Instead, decide which of the factors is *most* important, and compute the ROI of just that. Then, if there is a tie, you can break the tie by comparing "delight." This is also easy to explain after the fact: *Features A, B, and C are equally effective at generating revenue, which is our most important goal. Among those, Feature B also increases customer delight more than the others, so we're implementing B.*

Add instead of multiply (only when attributes are equally important)
Although it is arguably a lack of strategic decision-making to claim that multiple attributes are all equally important, it can make sense if you are scheduling secondary features. That is, suppose you've already scheduled the most important strategic items, and now you're "filling in" with "quick wins" where you truly don't care in what manner each is a "win." Here you add the values instead of multiply. This way, a huge ROI in one area wins (but we don't care which area). Getting *some* value in two

areas, is not as good as getting a *lot* of value in *one* area. This final point is satisfied when you use Fermi values, but traditional rubrics get it wrong.

BREAKING TIES

The Fermi-style rubric has another interesting property: **There are only a few unique values of ROI.** In the "revenue" example above, there are only 3 realistic numbers for revenue, and only 3 values of effort, so only 9 unique combinations of ROI.

The good news is, the computation clearly separates the best from the rest. The bad news is, you can easily make ties. In the "presentation" example earlier, in real life we had two topics in clear first and second place, but a three-way tie for third. We didn't have time for all five topics, so we had to break the tie.

These ties are not a failing of the system. Just the opposite: It means we are able to identify ties, rather than allowing noise to trick us into believing one is the "winner."

Still, we have to break the tie. There are several ways to do it, while preserving the advantages of this method.

Adding precision or intentional bias

It's tempting to add "precision" to your input values, and indeed that might break the tie. But beware of falling back into the accuracy problem and the explanatory deficiency of typical rubrics.

You should add precision only if:

"I remind you that I recommended against upping our game while simultaneously taking it to the next level."

credit 196

1. You are certain the precision is accurate, or
2. You are creating an intentional bias, e.g. to over-weight one factor relative to another.

There's a mathematical trap in option (2). It's tempting to "weight" a factor by multiplying it by a constant, e.g. "we'll double this score, so it counts twice as much as other scores." However, this actually doesn't do anything at all, because it just means all totals for all items are multiplied by 2; this doesn't change the ordering, and doesn't break ties.

Instead, what matters are the sizes of the *intervals between* the choices. That is, if scoring for this item was originally `1, 10, 100`, you could *reduce* how sensitive the total is to this value by diminishing the differences between values, e.g. by going up by powers of three instead of powers of ten: `1, 3, 9`.

You also don't need to keep the spacing regular. You might rank more options as a `0` to effectively "filter out" things that don't meet a desired standard. You might decide that the two options at the top

of the scale, that originally were `100` and `1000`, should be more like `800` and `1000`, reflecting a bias that "both options have essentially the same value, one just slightly more."

The run-off

Given a set of items that we've agreed are "equal in ROI," pick a new dimension to break the tie.

Time-estimates are no longer part of the process. We already agreed these are efficient in terms of impact-versus-time, so now we focus only on some additional dimension of value. This simplifies the process, which is not only nice for decision-making, but also for explaining the decision.

If you don't want to evaluate the items along a new dimension, you could eliminate some of the existing dimensions. For example, in the "presentation" case, we decided that, all else being equal, we would rather select topics where we have a really interesting insight, rather than topics that are great at selling the product. Therefore, we performed a run-off using "insight × reach," ignoring the other values.

To determine which is the most important dimension to maximize, using this guiding question:

> If we maximize this dimension in the next few months, then even if we get nothing else accomplished, it will still be a success.
>
> Conversely, if we moved other needles, but this dimension remained unchanged, it will have been a failure.

The human factor

A different way to break the tie is to include something else that you value, but that shouldn't be used to make the primary decision.

Since the primary decision is a tie, we're "allowed" to indulge ourselves in something else, even something unconnected to "business metrics" in the usual sense. Because we've already maximized a business metric, you could break ties using something completely different:

Team Excitement

Whether because it's fun, interesting, or the team is just tired of hearing certain customer feedback and really wants to do something about it, "excitement" is an excellent, under-appreciated attribute. Excited teams are happier, feel more fulfilled, feel more listened-to, and work harder and better. It is the elusive "win-win." (p. 627)

Since "excitement" is an ambiguous term, one way to convert to a Fermi value is the following:

Fermi Excitement	Definition
0	No one is excited to work on it.
1	One person is excited to work on it.
10	About half the team is excited to work on it.
100	All, or all but one, is excited to work on it.

Confidence

Confidence is hard to measure, but you know it when you feel it (or don't feel it). It might not be correlated with risk—just because you are confident, doesn't mean you are correct. Still, faced with two choices, where you're sure you know how to execute the first, but with the second you're full of doubt, it's logical to pick the first.

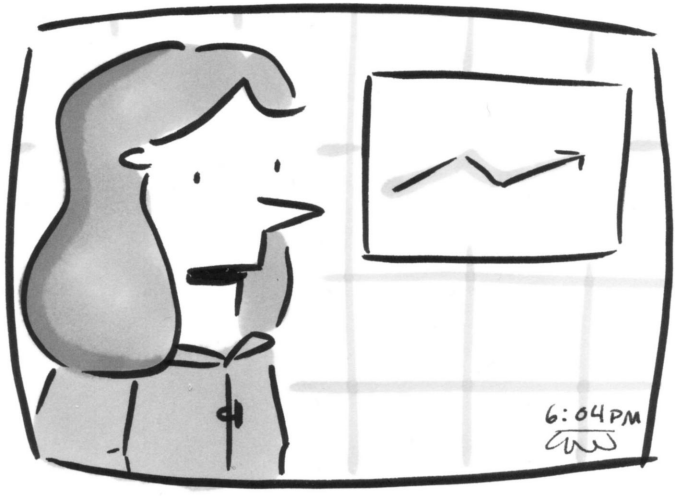
Since "confidence" is an ambiguous term, some people use some sort of risk percentage, e.g. "I'm 60% sure we can do this." But pinning down a number is suspicious (p. 997), and debates don't seem fruitful. One way to convert to a Fermi value is the following:

Fermi Confidence	Definition
0	There's no way we can pull this off.
1	I mean we can try, but don't be surprised when it fails.
10	We can probably do this; we'll be surprised if it goes very wrong.
100	This is completely within our domain of expertise, we've successfully executed something similar in the past, and nothing in the task-breakdown is uncertain.

Good luck!

Chapter 10:
Navigating the unpredictability of everything

PREDICTING MARKETS · PREDICTING SALES
PREDICTING WINNERS · PREDICTING PRODUCT
THE SOLUTION · STRATEGIES



"Reaction to the news was mixed, largely because we asked more than one person."

PREDICTING MARKETS

Analysts at Goldman Sachs spend their whole life learning advanced mathematics, building sophisticated models of complex corners of the world, and are financially incentivized to predict the future accurately, because Goldman Sachs makes billions of dollars if they can be 10% more correct than the next firm (who also employs brilliant analysts).

So, how accurately did they predict economic metrics within their area of expertise? They got it very wrong, for 25 years, often not even directionally correct, in things like T-Bill rates (Figure 1) and Oil receipts (Figure 2).

But maybe the macro economy is too hard to predict, even though these sorts things should be in the domain where we can leverage the "wisdom of the crowd (p. 931)." It is "chaotic," we are told, in the mathematical sense that small changes in inputs result in large, unpredictable changes in outputs. The "Butterfly Effect"[199] and all that.

When we create strategies, we're told to "skate where the puck is going." That means predicting the future, such as macro economic trends, and the trends of our industry and immediate competitive markets.

But if the smartest, most motivated experts can't do that, why do we think we can?

Let's try something closer to what corporate strategists must do: Predict the future for products in well-understood industries.

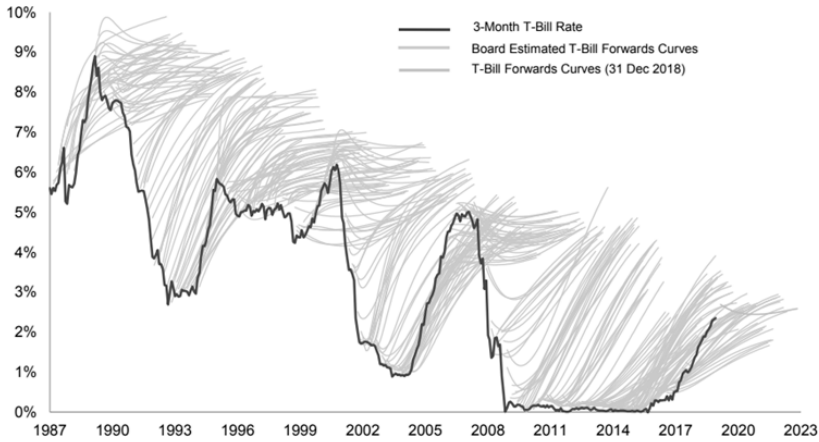


Figure 1: The thick line is the actual value of the metric; thin lines are quarterly predictions of how the thick line *will* move, from each starting-point.

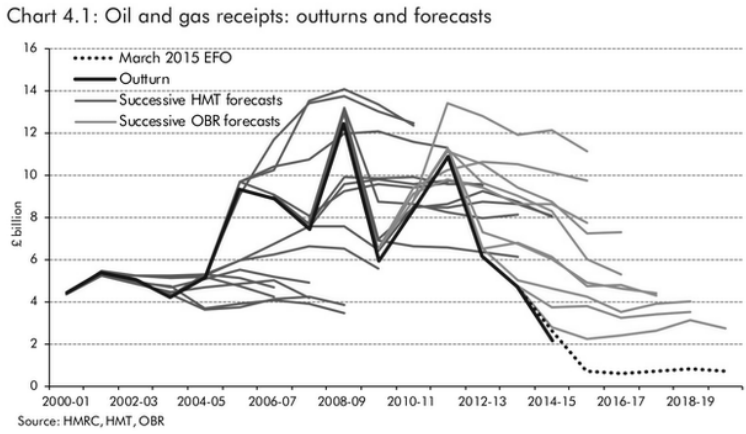


Figure 2

credit[200]

"So you believe that when you flap your wings on one side of the planet, all kinds of crazy things are caused on the other side? That sounds like a lot of responsibility."

PREDICTING SALES

> Researchers at McKinsey & Company, a consulting firm, studied the sales forecasts for new drugs made by analysts at brokerage firms. The data included more than 1,700 individual forecasts on 260 drugs over a recent ten-year span. ... The average error was large. **Almost two-thirds of the estimates missed the peak revenue amount by 40 percent or more.** Further, **forecasts for follow-on drugs were no better** than the first launches within a therapeutic class.
> –Michael Mauboussin & Dan Callahan in Total Addressable Market,[201] Credit Suisse 2015 *(my emphasis)*

Once again, highly incentivized, highly trained analysts, within their area of expertise, within a well-understood, highly-regulated industry, mostly get it really wrong, whether predicting sales of a new drug, or predicting sales of a new competitor to an existing drug.

Mauboussin and Callahan suggest that one way to increase the accuracy of predictions is to "use base rates as a reality check." That is, use the industry average as the starting point for your prediction.

Does this method work? They illustrate:

> Here's an example of how the base rate approach can figure into your judgment of TAM. During a conference call in February 2015, Elon Musk, the chairman and chief executive officer (CEO) of Tesla Motors, suggested the company might be able to achieve a **50 percent compound annual growth rate of sales** for the next decade ...
>
> ... The base rate method ... simply asks: "What happened to other companies when they were in a comparable position?" [Figure 3] shows the distribution of 10-year sales growth rates for more than **1,200 instances of companies of a similar size as Tesla is now**, measured by sales. The average growth rate, adjusted for inflation, is **less than three percent**, with a standard deviation below eight percent. Further, **no companies achieved a compound annual growth rate (CAGR) in excess of 40 percent** ... We have placed a star at the growth rate Musk mentions.
> —Michael Mauboussin & Dan Callahan in *Total Addressable Market*, Credit Suisse 2015 *(my emphasis)*

As we see from Figure 3, Musk's call is completely outrageous when using the base-rate method.

It is 2023 as of this writing, so we can now evaluate Musk's call, and see whether the base-rate method was correct (Figure 4).

So, the base-rate method is suspicious, and the analysts are suspicious. Is anything not suspicious?

When we build strategies, we're predicting what sorts of products are going to sell, often inside a fast-changing market, selling to customers with fickle desires in an ever-changing competitive landscape. Drugs that address a well-known disease with well-known rates of incidence and trends, with well-known competition, are easy in com-

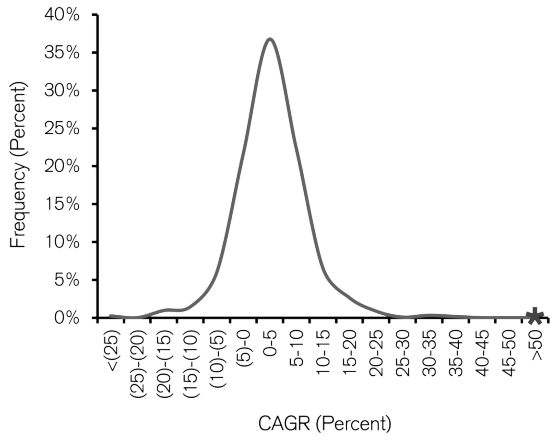


Figure 3: 10-year growth rates for companies with $6B-$13B of sales.
Source: Michael J. Mauboussin and Dan Callahan, "The Base Rate Book – Sales Growth," Credit Suisse Global Financial Strategies, May 4, 2015.

parison. Easy, but already almost impossible even by the experts. And using base-rates doesn't give us much confidence either.

So why do we think *we* can do it?

PREDICTING WINNERS

Chess Grandmasters spend far more than 10,000 hours reaching an elite, unbelievable level of skill. They instantly pattern-match board positions against thousands of games they've memorized. They have an intuitive sense of how the future might unfold, without actually thinking through every possibility. They understand the relative strength of the best players, because they've studied their games for years. So when they guess who is going to win the 85th annual Tata Steel

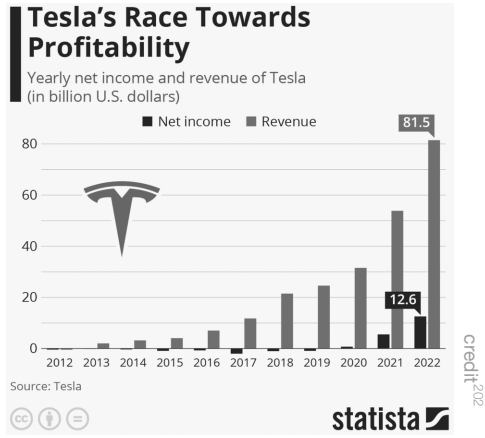


Figure 4: Actual Tesla revenue CAGR since 2015 is 55% — even more than Musk's 2015 claim.

tournament—arguably the most prestigious tournament on the annual circuit—it is more than just "sports betting." It's betting by deeply educated experts. How did their predictions go? We can see in Figure 5.

The "base rate" prediction is that Magnus Carlsen—the world #1 in all three major categories of chess, arguably the best player to have ever lived—should win, and indeed he was given the highest probability of winning before the tournament began, even up to round 4.

But the world doesn't necessarily unfold around the base-rate. Nordirbek Abdusattorov—an 18-year-old junior player rated more than 100 Elo points lower than Carlsen (an enormous gap)—was given such a small sliver of a chance at the start that it's hard to even read the number. He lead the tournament standings until the very last game. He also beat Carlsen in round 5.

And Giri was never given more than a 30% chance; usually 10% or less. He won the entire thing.

The same thing happened during the last FIFA World Cup. Argentina—the eventual winner—was never given much of a chance; even in the quarter-finals the bets were on France (Figure 6).

Daily Mail, Tuesday, December 5, 2000 Page 33

Internet 'may be just a passing fad as millions give up on it'

THE Internet may be only a passing fad for many users, according to a report.

Researchers found that millions were turning their backs on the world wide web, frustrated by its limitations and unwilling to pay high access charges.

They say that e-mail, far from replacing other forms of communication, is adding to an overload of information.

Experts from the Virtual Society project, which published the report, say predictions that the Internet would revolutionise the way society works have proved wildly inaccurate.

Many teenagers are using the Internet less now than previously, they conclude, and the future of online shopping is limited. Steve

By James Chapman
Science Correspondent

Net loss: Two million Britons have logged off the Internet

Woolgar, director of the society, said: "We are often presented with a picture of burgeoning Internet use, but this is already at drop-off and saturation among users.

"Teenagers' use of the Internet has declined. They were energised by what you can do on the Net but they have been through all that and then realised there is more to life in the real world and gone back to it."

The project, sponsored by the Economic and Social Research Council, gathered together research by 25 universities across Europe and the US.

It estimated that in Britain alone there could be more than two million people who regularly used the Internet but had now given up.

Analysts say some

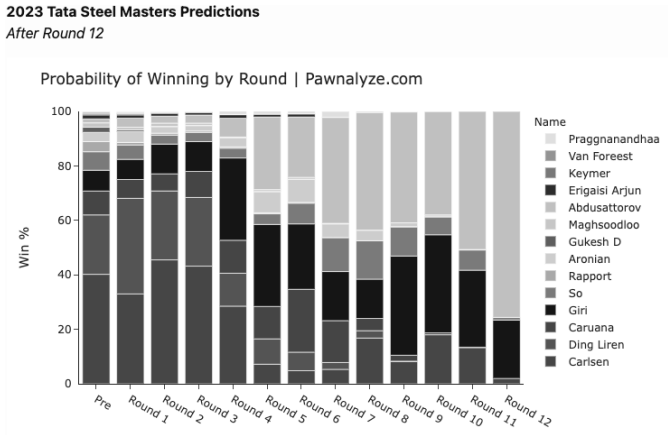
EXCLUSIVE!

Daily Mail, from the year 2000.
Choice excerpts: "e-mail [is] far from replacing other forms of communication" // "the future of online shopping is limited" // "teenagers' use of the Internet has declined … they've been through all that and then realized there is more to life in the real world and gone back to it." // And who generate these non-sense predictions? "Experts from the Virtual Society Project" comprised of "research from 25 universities across Europe and the US."

When we build strategies, we're predicting the competitive market and which products will be winners. Chess and soccer games have a higher variance than companies, but analyzing sales funnels sometimes feels a bit like predicting Tata Steel. How do you build a strategy around this level of unpredictability?

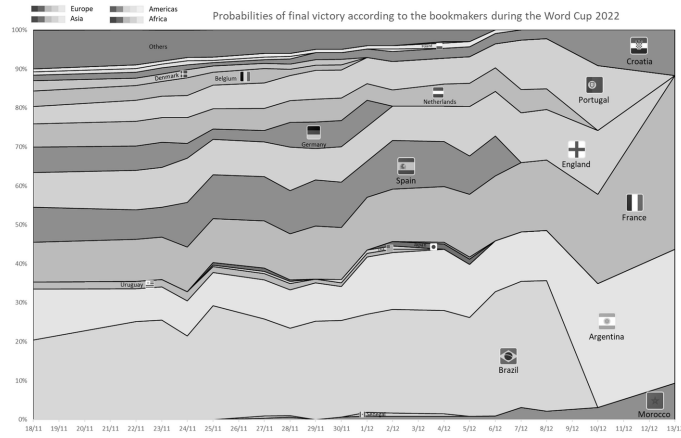
PREDICTING PRODUCT

Stewart Butterfield always wanted to make a game. So, in 2002, he did (Figure 7).



credit 203

Figure 5: Notice Giri — initially tiny, then growing, then shrinking by round 8, then growing again, then being crushed by Abdusattorov on round 12.
Giri won the tournament on round 13.



credit 204

Figure 6: FIFA World Cup 2022 prediction-evolution over time

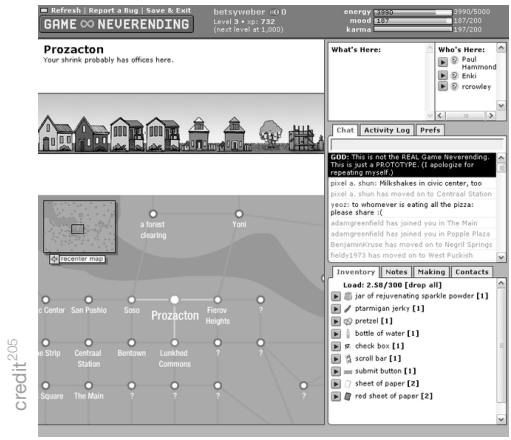


Figure 7: Game Neverending: An in-browser multi-player on-line game "with no way to win, nor any definition of success." (Like some companies I know)

It failed, but it had some interesting features. Alpha testers liked that they could share game objects by dragging them into chat windows. So, the engineers created an online application for real-time chat with image-sharing.

Stewart Butterfield is the founder of Slack, so you might guess that this "better chat" idea became Slack, but no.

The chat application only worked in real-time—your pictures didn't stick around when you closed the app. And this was fatal because it turns out people were interested in the *sharing* part more than the *real-time* part. So in yet another upheaval they rewrote the Flash application as a regular website, and Flickr was born. By 2008 it had become the largest photo-sharing site in the world with 3 billion photos and 5,000 more uploaded every minute.

Yahoo bought Flickr for $25M in 2005; Butterfield left three years later. Now that he had money in his pocket, he was free to go back and do what he always wanted to do, which was to make a game. So, in 2009, he did (Figure 8).

credit[206]

Figure 8

There's a whole story[207] about their dreams of a massively-multiplayer experience, with APIs so others would build even more things on top of the game.

But it doesn't matter, because once again it was a failure, and shut down.[208] But once again there was a piece of the game that people really liked, and once again it was the chat system. Butterfield pivoted the company with a new mission, and a new name that belied the mission: SLACK, the Searchable Log of All Conversation and Knowledge.

Still not "chat," but rather a "searchable log of knowledge." It's too hard, Butterfield insisted, for corporate denizens to search Google docs, and email, and chat sessions, and intranets, and knowledge bases, and support channels, and sales logs. Slack brings all that content together into a single omni-search, thus solving a knowledge-management problem common to all companies.

Butterfield was certain that merely "building yet another chat system" was a bad strategy, whereas "transforming communications" was a good strategy:

> We are unlikely to be able to sell "a group chat system" very well: there are just not enough people shopping for group chat.

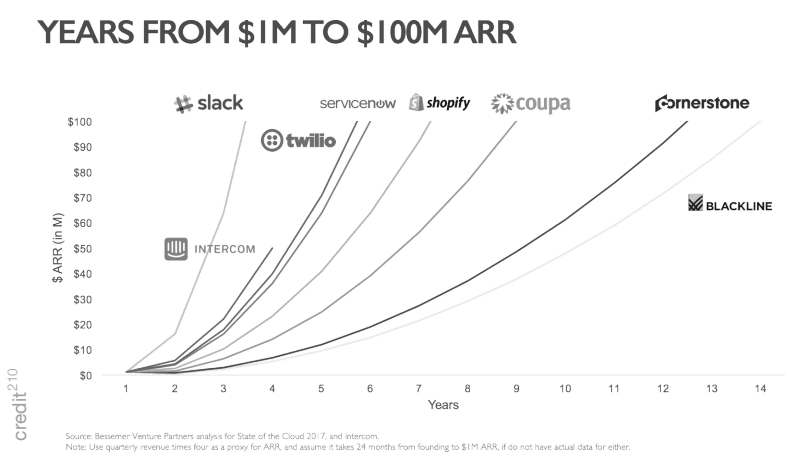


Figure 9

> That's why what we're selling is organizational transformation.
> —Stewart Butterfield, Slack founder, in 2014[209]

Once again, great theory, great mission, great strategy... and it didn't work. Slack was, and is, yet another chat system.

Fortunately for him, Butterfield was also wrong that "just chat" wouldn't sell very well. Slack was one of the fastest-growing companies in history (Figure 9).

What it is about people not wanting to build chat? WhatsApp has a parallel story. Initially it was just a way to post a public status message, so friends could see what everyone is up to, hence "WhatsApp" (like "What's Up"). No one cared. Then the iPhone launched push-notifications, so they added a feature to alert you when a friend's status *changed*. People started abusing this as a crude form of group-chat. So they added group-chat as a proper feature. The correct strategy was simple: SMS costs money for most people in the world, so a winning product strategy is "SMS, but free." That strategy worked (Figure 10).

When we build strategies, we're predicting how customers will receive our product, how it will solve problems or be delightful. And yet

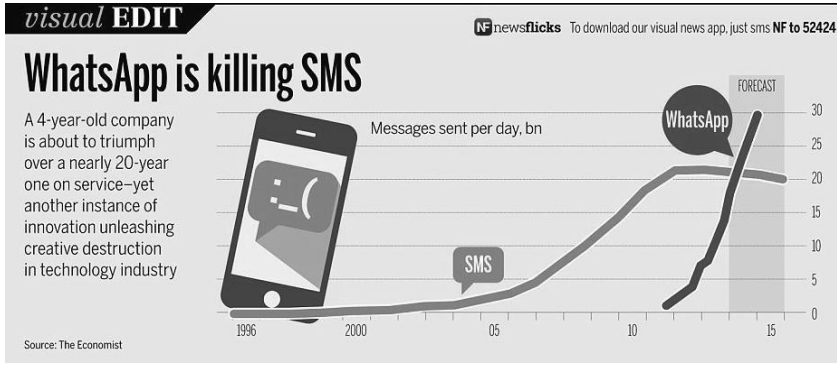


Figure 10

so often the idea we start with isn't the idea that wins. Even once Slack became "Slack" and not "Glitch," the strategy was still wrong. How can we ever be right?

THE SOLUTION

The Slack and WhatsApp stories of unpredictable success have at least two things in common:

1. They had a strategy—a philosophy of what would be entertaining or useful, a design sense of what would be delightful, brand new ideas for features and user interactions that people loved. They didn't just throw random things at the wall.
2. They went where the customers took them—if not a game, then sharing images; if not another game, then better chat; if not corporate omni-search, then chat with APIs, if not status updates, then free SMS.

A strategy is required, even when it's wrong.
A strategy gives you a direction. A direction creates something interesting. Something interesting might be used by zealous early fans, even if it ends up uncovering the answer rather than being the answer.

If Game Neverending wasn't delightful enough to attract alpha testers, there would never have been a reboot into Flickr; if Glitch didn't have 150,000 users at its peak, there wouldn't be the idea for Slack; if Slack didn't have enormous ambitions, it would not have received VC funding (for the same reason Butterfield himself gave for why "yet another chat" isn't valuable).

A corollary: Making a decision and moving forward is often more effective than extended deliberation about the decision. Deliberation assumes we know how to reason about the future, but even experts aren't good at that. Making decisions and gaining experience is how to find the right answers, if the organization is introspective enough to also face the truth (p. 657) when it turns out the original strategy is incorrect.

Dispense with the idea that there is One True Solution to the puzzles, and that the way to get there is to gather enough information. While "iterate fast, learn fast" is good advice (p. 449), it still doesn't mean you're iterating towards the One True Solution. It means you're taking a path towards *something*, which will turn out to be an integration between your own creation and the swirling reality outside of you.

> *"More than any other time in history, mankind faces a crossroads. One path leads to despair and utter hopelessness. The other, to total extinction. Let us pray we have the wisdom to choose correctly."*
>
> *–Woody Allen*

The customer (behavior) is always (directionally) right.

There is a contradiction in these two points. Point (1) is to have a firm strategy, yet point (2) is to go wherever the customers are, even if that violates (1). If you just do "whatever customers want," how is that a strategy?

The answer is: Customers are where you discover how to upgrade your strategy. Since you know your initial strategy is wrong, following customers' lead is how you correct it.

When a customer loves one feature especially, or asks for adjacent features, those requests themselves are not your new strategy; the intent behind them might be. So "give me push notifications for my status updates" is a feature request, but the *reason* they want that is to circumvent SMS, and incorporating *that* idea is the strategy upgrade.

Perhaps the most interesting signal is when **the customer abuses your product to accomplish something else**. The desire for that "something else" is so great, they're willing to use the wrong tool to get it done. Not only exemplified by WhatsApp users abusing "my status" to be "group chat," but also in my own experience getting Smart Bear to start growing:

Our first product (Code Historian) let you visualize the history of your codebase, which seemed cool to me but few people paid for it (Figure 11).

But people abused it to enable peer review. We were getting feature requests like: "Let me package up what's on my screen and send it to someone else," and "Let me write directly on a line of code and send it back." (Figure 12)

A new product was born (Code Collaborator) which within a few years represented 97% of the revenue of the company. "Abuse" is a strong, positive signal for customer demand, and worthy of a change in strategy.

So yes, the future is unpredictable. But that doesn't absolve you from creating a strategy, and it certainly doesn't mean you should accept that life is a random walk, with no way to bias results in your favor. In fact, sometimes strategies are correct right from the start, like

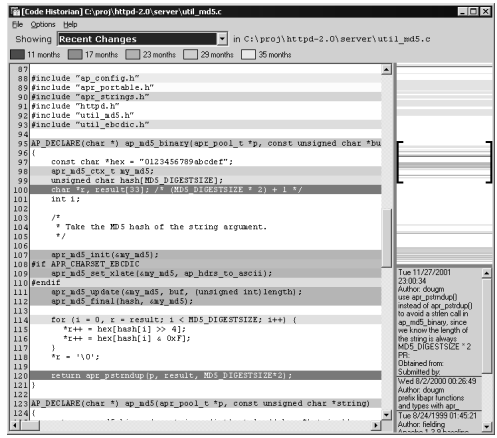


Figure 11: I realize how old this screenshot looks, but it also means these insights have withstood the test of time.

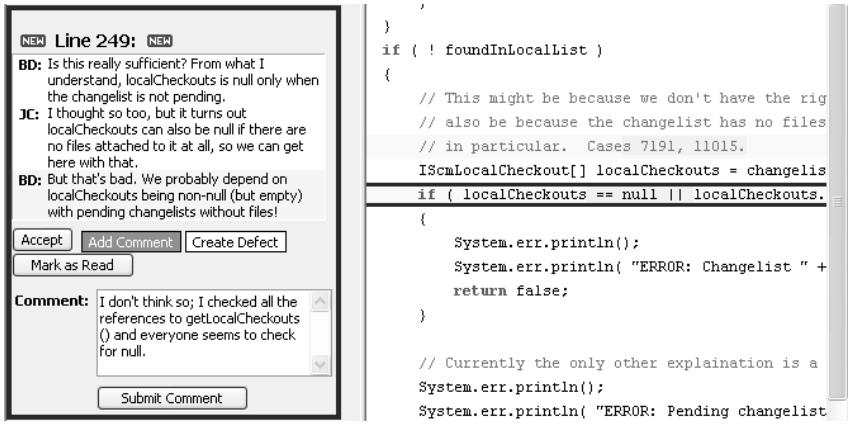


Figure 12: Twenty years later, this is a standard feature of many developer tools, but at the time it was an innovation that fueled a multi-million-dollar bootstrapped company.

it was with Google, LinkedIn, Amazon, Apple, and my latest company WP Engine.*

You need a strategy, a fixation like Butterfield has for a certain type of game, a galvanizing reason for everyone to act together with a common purpose. And yet the strategy is always under suspicion, always updating, always reacting to unpredictable realities.

How, exactly, do you do this?

STRATEGIES THAT DEFEAT UNPREDICTABILITY

While it might not be possible to predict the future, it *is* possible for a strategy to side-step unpredictability through these mechanisms:

Build a moat

Covered in this companion article (p. 761), a "moat" is a long-term, durable competitive advantage. A structural advantage that others cannot disrupt, lasting for years, is resilient to the volatility of your competitive market.

Have more than one way to succeed

If multiple things all have to go right for a plan to succeed, it probably won't succeed. That sort of plan tacitly assumes we can predict the future of many different components; surely a bad bet.

* Although one reason WP Engine was correct at launch was because I used a system to discover what was important (p. 239) before building it.

This is why startups are difficult in the first place: We have to have an insight *and* build a compelling product *and* be able to reach target customers *and* do that cost-effectively *and* at a price customers will accept *and* be better than the competition for some segment *and* not lose a key team member early on *and* not have global economic failure. Like this (p. 71).

A resilient plan is built of "or" not "and." (p. 1277) We could reach customers through social media *or* paid advertisement *or* influencer marketing *or* channel sales. A product that at least *could* be sold in all those ways is more likely to succeed than a product which can only be sold a single way.

Optionality defeats unpredictability:

- A product that is very low cost to create has many options for pricing (p. 515), and therefore more likely to find effective pricing.
- A product in a large, growing market has many niches and personas and channels to potentially target, and therefore it's more likely you'll find some combination that works.
- Some indie developers build multiple small products, then pour their effort into whichever one happens to take off.

More examples are given in Nassim Taleb's concept of Antifragility.[212] In his words:

> If you "have optionality," you don't have much need for what is commonly called intelligence, knowledge, insight, skills, and these complicated things that take place in our brain cells. For you don't have to be right that often. All you need is the wisdom to not do unintelligent things to hurt yourself (some acts of omission) and recognize favorable outcomes when they occur.
> —Nassim Taleb, *Antifragile: Things That Gain from Disorder*[213] (2014)

Leverage your assets

"Strike where you are strong and the enemy is weak" said Sun Tzu 2,500 years ago, echoing into strategy-design from the SWOT of the 1970s to modern agile frameworks like the "Means" in Effectuation.[214]

This is obvious but often forgotten by entrepreneurs who believe they have found lucrative opportunities with amazing product ideas, but which fall outside of their sphere of competency, where the enemy is strong and they are weak. Any venture is likely to fail for many—unpredictable—reasons; the very least you can do, is leverage (p. 543) your capabilities, knowledge, network, (professional) friends, and accumulated assets. That might mean building a very different kind of company or product or target market; in fact that's exactly the point. When you pick the battle you are best-suited to win, you have a higher chance of winning regardless of the unpredictability of the world around you.

Intentionally reactive

If you cannot predict the future, one option is to spend no time whatsoever trying to plan. Just react to what you see now, and use your judgement and a long-term strategy to solve the immediate opportunities while advancing a long-term agenda.

We know, we know... iterate fast, learn fast, get to the right answer fast. We know that, we say that, but then we take four months to deliver one feature, or take 18 months before we're "ready" to launch. Truly embracing and living the idea of constant delivery, constant feedback, constant learning loops, constant adjustment of hypothesis, quick decision-making, quick updating of prior decisions, no ego tied up with who was right or wrong about what, treating all doors as two-way doors,[215] more than lip-service but encoded into the DNA of the organization, can overcome unpredictable barriers. There was no real prediction to begin with, just a constant stream of hypotheses, contin-

uously updated, possibly using frameworks and processes specifically designed for this mindset.*

Hedged bets

If you buy 30-year life insurance and live for 60 more years, you will make far more money than the insurance costs, so you come out ahead. If you buy 30-year life insurance and die in 2 years, your family will receive far more money than the insurance costs, so you come out ahead. Thus "insurance" is a way to always come out ahead, without predicting the future.

The catch: You've reduced your maximum upside. In the 60-year case, you paid insurance premiums for 30 years while receiving no money in return. On balance, however, this "tax" is worth it, because you're trading slightly less maximum upside for predictable, net-positive outcomes.

Strategies can also "buy insurance," i.e. come out ahead either way, albeit with a tax. Examples:

- **Multiple vendors for same service.**** Shift workloads based on price and performance. *(How would your price-negotiations go, if you were in this position of power?)*
- **Multiple brands** (whether created or acquired). Each finds a different niche, some will fail (p. 1261), some will take off. Common with acquisitions of growing companies, or the "house of brands" business model, or the "holding company" investment vehicle.

---

* e.g. building in public; the OODA loop;[216] Iterative Hypothesis Testing (p. 239); Continuous Discovery,[217] Continuous Delivery[218] (not to be confused with Continuous Deployment, in which deployment is automated but isn't happening many times per day)

** Wrap 3rd-party APIs in a proxy layer. Use multiple public clouds. Process online transactions with more than one party. Keep your cash in more than one bank.

"You huff and puff, we move in with our brother, and we all split the insurance money."

credit[219]

- **Multiple simultaneous solutions**. To create iOS, Apple had[220] the iPod team try to expand their OS to be more capable, while simultaneously having the Macintosh team try to reduce their OS to fit on a mobile device. One result was in-fighting, demonstrating that "cost" can be personal. But the result was that they selected the correct way forward for the iPhone.
- **Redundant systems**. City power, and a generator. Multiple servers in different geographic regions. Teams, not solo engineers. Lying dials in airplanes (p. 1267). Redundancy is a cost, but the result is more predictable operation. As the military saying goes, "Two is one, one is none."
- **Disrupt thyself**. Create new products that disrupt your existing products. The quintessential example is Amazon launching the Kindle; Bezos instructed[221] the newly-annointed Kindle leader—who was the leader over their book-selling business—that "Effective tomorrow, your job is to kill your old business with a Kindle."

Whether the future is in physical books, electronic books, or both, Amazon prospers.

Extreme novelty

Zappos, Airbnb, Uber, SpaceX, Tesla, Bitcoin, OpenAI, all took enormous risk by creating a new category that everyone else thought was impossible, even after they heard the plan. On one hand, this is the definition of unpredictability. On the other hand, they avoided the unpredictability that comes from existing competition in existing markets. It doesn't matter what's happening at Amazon if you can return shoes for free even after 364 days; it doesn't matter how the hotel industry is shifting when you're selling a different experience; it doesn't matter what Ford and Toyota are doing with electric vehicles if you're changing the fundamental technology without the constraints of existing supply lines, dealer distribution rules, and ties to the oil and gas industry.

If everything is unpredictable anyway, why not earn strategic advantage: No direct competition, exciting place for top-talent to join, immense upside.

That upside does need to be immense for the risk to be rational. If it is, this a strategy for trading into a better set of uncertainties.

Form coalitions

The more massive the object, the more it resists being moved by external forces.

Price-collusion* removes the uncertainty of market pricing. Peace treaties prevent some wars. Open source projects attract more talent and advocates than any one organization could afford. Industry stan-

* This is unethical and illegal in many countries; don't do it. It is illegal because it works, which makes it a good example.

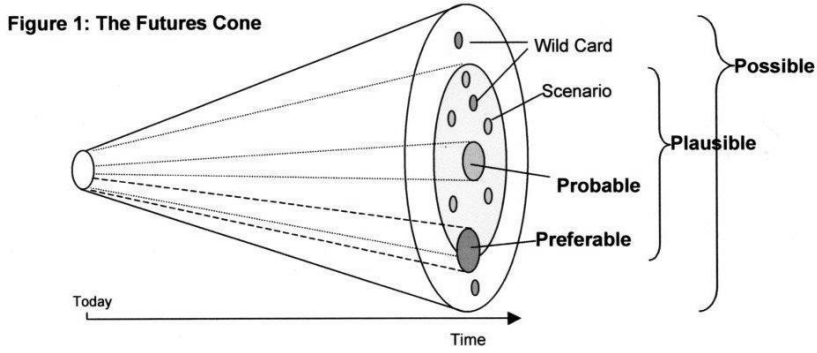
dards reduce the risk that core technology or protocols change, allowing members to build systems on top of those standard for decades, not having to "hedge." Good things happen when we all agree to use HTTP, HTML, and SSL.

> *"If you want to go fast, go alone. If you want to go far, go together"*
>
> *–African proverb*

Expand the scope of prediction

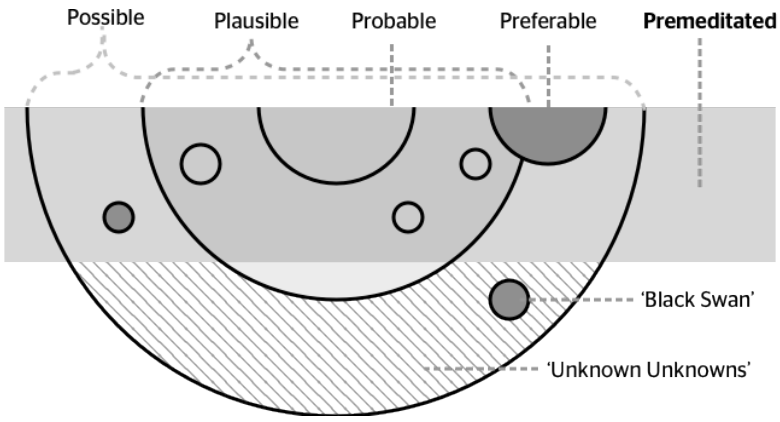
Predicting exactly what will happen is folly, but mapping possible futures can be illuminating (Figure 13).

Plotting possibilities that you believe are improbable, helps you recognize that perhaps they're not quite as improbable, helps you think of solutions that mitigate plausible challenges,* and help you recognize if an "improbable" thing is in fact happening.



credit[222]

Figure 13: "Futures Cone" from Voros 2003,[223] expanded by [](https://sjef.nu/theory-of-change-and-the-futures-cone/) Sjef van Gaalen in 2016



credit[224]

Figure 14

Of course you're still ignorant not only of probabilities, but of eventualities that you never predicted. Gaalen acknowledges this by further expanding the framework (Figure 14).

While this doesn't mean you suddenly can predict the future, it might mean you're paying more attention to what's actually happening, allowing you to react faster, and thus to *manage* the unpredictability better.

Stay simple

The more tasks have to be broken-down, the more dependency arrows are drawn, the more intricate the analysis, the less believable the project estimate is. Complexity breeds unpredictability.

---

* The "pre-mortem" is an increasingly popular workshop for accomplishing this, in which you brainstorm answers to the following question: "It's 12 months from now, and the project is a disaster. What went wrong?" The idea isn't to solve everything you can imagine, but rather to pick a few things to intentionally mitigate, and intentionally leave the door open for mitigating other things if they come to pass.

The inverse is that simplicity is predictable. While not strictly true, is it true that simple things can be more predictable. A simple product, with a simple value proposition, in a simple market, has fewer dependencies that require prediction, and thus are more likely to succeed.

Bet on what will not change

Made famous by Jeff Bezos, it is difficult to predict how the future will change, but it can be easy to predict the ways the future will *not* change. In Amazon's case, he cites "low prices" and "fast delivery" as two of those things; in ten years, people will still want that. Therefore, Amazon can (and does) invest billions of dollars to achieve those results.

In the case of WhatsApp, consumers want to chat and don't want to pay; that was true ten years ago, it's still true today, and it's a good bet that it will be true ten years from now. In the case of WP Engine, the company I founded 15 years ago, people wanted websites to be fast, stay fast even when they get a lot of traffic, and be secure; they still do, and I'm sure they still will in another 15 years.

Gukesh Dommaraju entered the 2024 Chess World Championship as the overwhelming favorite, with pundits unified in predicting his victory. The only debate was how quickly and decisively he would defeat Ding Liren, the reigning champion who had been underperforming for years.

Then Ding won the first game with the black pieces—a remarkable feat, since black is typically played for a draw. The pundits immediately reversed course, declaring the match 50-50 and Ding a force to be reckoned with.

By the 14th and final regulation game, each player had won twice, with the rest drawn. Five hours into that decisive game, commentators could see it was heading for a "dead draw"—meaning that even lower-rated players would have no trouble avoiding a loss, but also unable to force a win. One pundit confidently declared in his livestream: "There's a 1% chance Ding doesn't draw this game."

Five minutes later, Ding blundered, Gukesh won, and became the youngest World Champion in history.

And so the lessons are:

- Have a strategy, even though the world is unpredictable.
- Decide quickly → get customer reactions quickly → learn quickly → make new decisions quickly.
- Upgrade the strategy by following customer behavior.
- Product "abuse" is a strong signal for updating the strategy.
- The strategy must include building a moat (p. 761) or two.
- The strategy must create optionality in how to succeed (or avoid failure) so that single failures aren't fatal.
- The strategy must leverage your existing strengths (p. 543), building a product for a market for which you are already well-suited.
- Either keep it simple, form coalitions, or do something completely novel.
- Bet on things that *won't* change, rather than predicting how things *will* change.
- Do your homework (p. 239), but don't stall in analysis paralysis. Map future possibilities, to mitigate possible things and to better notice when your assumptions turn out to be incorrect.

That's how you win in an unpredictable world.

Many thanks to Adam Brock,[225] Amy Hoy,[226] Daniel Zarick,[227] Darla Cohen, Derrick Wolbert,[228] John James Jacoby,[229] KimSia Sim,[230] Nitin Punjabi,[231] Paul Huggins,[232] Prasanna Krishnamoorthy,[233] Rhys Jeffery,[234] Seth Chasin,[235] and Tony Meijer[236] for contributing their insights to early drafts.

Chapter 11: Rocks, Pebbles, Sand: How to implement in practice

THREE MINDSETS · ROCKS MAXIMIZE IMPACT
SAND MAXIMIZES THROUGHPUT
PEBBLES MAXIMIZE ROI · SPRINT-PLANNING

You know the geology-in-a-jar lesson from Stephen Covey:[237] Schedule big things first, otherwise you run out of time (Figure 1 & Figure 2).

A common mistake is to think this applies only to the *size* of the work. That is, "Rocks" means "stuff that takes a few quarters," "Pebbles" means "a few sprints," and "Sand" means "less than a sprint."

This misses the most important point of work-ordering: It's about maximizing *impact* by not allowing the easy or urgent things to crowd out the strategic things that take years to unfold but are more important than everything else combined. A thousand "quick wins" do not create durable advantages or fulfill a long-term vision.

Another mistake is to think that the previous paragraph is the end of the story. "Schedule revenue-growth stuff, then maintenance updates, got it." No. Each type of work requires different prioritization

credit[238]

Figure 1: If you do little things first, there's no time for big things.

credit[239]

Figure 2: If you do big things first, you can fit in smaller things.

frameworks, has different goals, and hide different traps that make you unwittingly ineffective.

If you pretend these differences don't exist, your team will be working hard and delivering lots of code-commits—the appearance of "productivity"—but they'll feel like they're not making progress fast enough, competition will start catching up, and they'll (correctly) complain that they can't see how their work is connected to the strategy.

The good news is: It does not take additional time to do it right. This is an instance of "smarter, not harder." You need the right frameworks.

THREE MINDSETS

A tabular summary is trite, but it's a handy reference:

	Rocks	**Pebbles**	**Sand**
Effort	≥3 Months	1-4 Sprints	≤1 Sprint
Maximize	Impact	ROI	Throughput
Outlook	Long-term	Short-term	Immediate
Scope	Strategic	Tactical	*any*
How Decide	Deliberate (p. 603)	Analytical (p. 171)	Intuitive
Role of Exec	Decider	Observer	*none*
Role of PM	Driver	Decider	Decider *(but engage devs)*
Beware	insufficient impact	over-estimating ROI	over-thinking / over-planning

Now we'll justify and explain how to use this to maximum effect, each stone in turn.

credit[240]

"No, I'm working from home today."

ROCKS MAXIMIZE IMPACT

Duration: 3-12 months

Rocks take the most time; let's call it 3-6 months. Long projects are not only expensive, they're also most likely to over-run, because they're the most complex, contain the most unknowns, and have the most dependencies. So it's really 3-12 months.

> **Hofstadter's Law**
> It always takes longer than you expect, even when you take Hofstadter's Law into account.

Well-oiled agile teams will point out they can hit earlier deadlines by adjusting scope and pushing less-urgent items past the deadline. That's wonderful for learning and customer-delight. But the leftover work still needs to be done, even if rearranged, so this doesn't change

the magnitude of the effort required to achieve the full effect of the idea. And anyway, there are good reasons[241] why we're consistently incapable of estimating big projects.

A team might complete only one Rock in a year—certainly no more than three—especially because other work also needs to get done. If you can do only one big thing this year, that thing had better be extraordinary.

Maximize impact

A Rock must deliver dramatic, measurable impact, not merely "incremental improvement." It must be *strategic*, meaning that it must attack the most important challenges you face (p. 1065), materially advancing the company down its unique path (p. 891) for winning its corner of the market, leveraging existing advantages to reduce risk and to forge a path that others cannot easily follow, and build new durable advantages (p. 543). This is where teams most often fall short: Not delivering enough impact to justify their investment (p. 867) of time.

Sadly, big projects not only over-run on time, but also often under-deliver on impact.* These sorts of predictions are famously inaccurate (p. 193). So it's even more imperative that we demand an enormous impact: That way, if we under-achieve, it was still worth the time.

Crucially, and perhaps controversially: **Do *not* maximize ROI.****
Your primary job is to execute your strategy to the fullest, spending the most-possible time on the most-impactful thing. If an idea is less impactful, yet also quicker to achieve, that is *not* the right choice. When things go worse than planned, that "less impact" turns into "incremental impact," and you cannot spend half a year on something so trivial.

---

* Fortunately, on occasion they can over-deliver by an order of magnitude or two; this is always the *post hoc* story of a successful company, the founders shaking their heads saying they never believed it would be *this* successful.

** ROI is "Return on Investment," computed as a measure of impact divided by a measure of effort, resulting in a measure of efficiency, i.e. "value per sprint." (Whatever "value" means.)

Beware: "Maximizing impact" is harder than you think

The most common problem is executing Rocks that aren't impactful enough. The Rock claims to "make a difference," but not *enough* difference, and after a few years, it feels like "we're not moving fast enough" or "why isn't revenue higher" even though the work from engineering is high-quality and stories are duly delivered every sprint.

Hofstadter's Law applies not only to the time-estimate, but to the impact-estimate, and thus to the height of the bar that need to set: It must be higher than you think, even when you take Hofstadter's Law into account. Over-shooting is the antidote to Hofstadter's Law.

Even if your ideas aren't good enough, you will be tempted to select the most impactful idea on the list and just do it. This is a mistake. It's such a common mistake, it is a cliché: "Good" is the enemy of "Great."[242] You have to face the truth (p. 657): Your biggest problem is a lack of a truly great idea, and you must solve *that* rather than embarking on a long, misguided journey. The team can do Pebbles and Sand in the meantime, thus staying productive, while also helping create and validate better ideas.

Deliberative decision-making process

Because you're committing so much of the team's life,* and because you have to be so confident that the Rock is strategic and impactful, you need to spend time on this decision up-front. This is not an "agile" decision, it's a strategic one. Once the direction is set, the big picture is clear, the mountain you want to climb is identified, *then* it is ideal to be "agile" in how you climb it. Execution details are never certain; backtracking is necessary. But if you're climbing up the wrong mountain in the first place, being "agile" doesn't help; the result is a

* Agilists argue that if you find yourself part-way through a failing project, you can just abort, because "that's agile." That's true, and that's smarter than plodding forward in a sunk-cost[243] refusal to face reality, but on a human level it is demoralizing and ruins trust to abruptly cancel a project a team has been laboring on for months, "because we're agile." Canceling is necessary, but not free, so you should act *as if* a Rock is a one-way door.[244]

team self-managing themselves into a mediocre, unfulfilling result. Although most decisions should be fast, sometimes they should be slow (p. 745); Rocks should be slow, or at least deliberate.

Use this framework to select the most impactful idea: Binstack: Making a maximal multi-dimensional decision (p. 603). This process enshrines "impact" as the highest priority, allows other dimensions to participate but neither confuse nor dominate the decision, and produces a pithy, clear explanation of the decision at the end.

If you're not coming up with good-enough ideas in the first place, try these prompts (p. 53). But also consider whether the real problem is that your strategy is too vague.

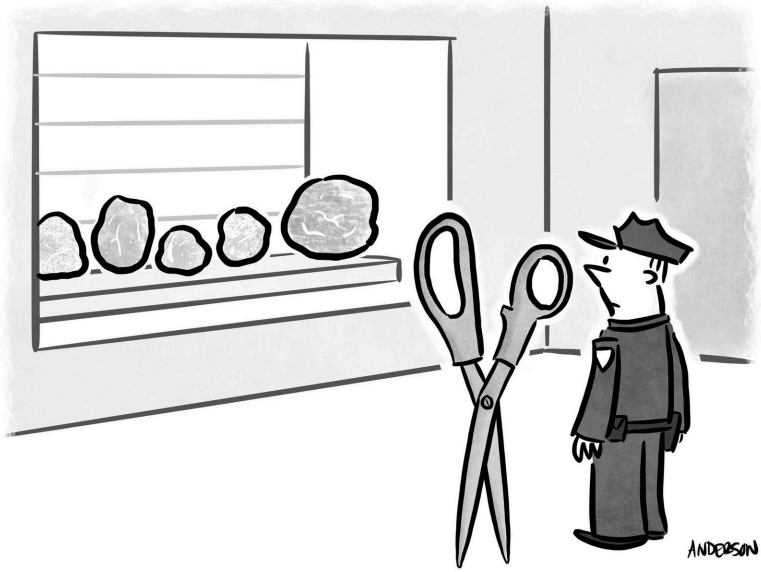
Execs decide, but ideally PMs are in command

Rocks materially advance the strategy, and executives* ultimately own the strategy. So, ultimately the final decision of what Rock to execute rests with the executive. In practice, however, the PM should be in command of the strategy and the ideas, driving the discussion and the decision. Ideally** the PM is actually in command (p. 413), and the executive is happy to play the role of coach and Devil's Advocate (p. 1017) during the decision process, and to sign off on a well-thought-out proposal.

---

* At a smaller company, this means whoever has the executive role, whoever is running all of Product, or R&D, or the CTO, or the CEO.

** We won't cover workplace dynamics in this article, so suffice to say that this is what "healthy" looks like, and the further reality is from ideal, the more one or both parties needs to change.



credit[245]

SAND MAXIMIZES THROUGHPUT

Because Pebbles are the Goldilocks of work-items, it's instructive to leap over them and solve for the smallest items, establishing bookends around the middle-child.

Duration: ≤1 sprint

Sand are items that don't need to be broken down. They're short, and typically "just need to be done" without discussion or ado. They can't take longer than a sprint; sometimes they take less than an hour.

There are a million little things that are individually unmeasurable but that add up to a significant impact. Great user interfaces require a hundred tweaks to attain greatness. High-quality error-handling requires esoteric corner-case unit-tests. High performance is often the result of innumerable optimizations. High-quality code means fixing myriad little bugs, some of which customers never experienced. Con-

tinuous, low-risk tech-debt reduction requires a hundred small refactorings. Security patches and library upgrades are mandatory maintenance. Documentation tweaks are helpful, low-risk, and should be done continuously. Great writing in general requires myriad tweaks, not grand organizational "pivots;" most edits to this document change a single sentence.

In bulk, Sand is mandatory for wonderful, high-quality software. When you use a piece of software and say, "Wow, this is really well-done," that's a result of Sand. While each grain is typically impossible to "measure," ignoring them means we will never attain greatness, even with the best strategy.

Maximize throughput

Sand has a material impact only when executed *en masse*, as a sort of opposite to "death by a thousand cuts." Therefore the goal is to maximize how many of them we fit into the interstitial spaces between the Rocks and Pebbles. The measure of success is throughput: How many we complete per sprint.

We cannot—and should not—try to measure or prioritize "impact." It's too small to measure.* It's enough to agree that fixing ten bugs this sprint is a great accomplishment, and that our customers and support techs will thank us for it.

Also, high-throughput is fulfilling and energizing for teams. It just feels good to cross lots of things off a list. Do not discount the importance of the *feeling* of productivity and usefulness.

Beware: Administrative overhead destroys throughput

Exactly because each grain of Sand takes little time, management processes will dramatically bloat the total time from conception to priori-

* It is conceivable that two days' work has a measurable impact on some metric (p. 645). When you happen across one of these four-leaf clovers, you obviously should do it, regardless of whether you consider it Sand (because it's fast) or a Pebble (because it has impact). The point is to grant yourself the grace for Sand to *not* have a measurable impact.

tization to completion. Administration is the biggest impediment to throughput, therefore PM's must ruthlessly fight against the natural urge to debate and arrange and carefully assess and estimate the work, instead of just going about the business of completing the work.

If you're using the same processes to prioritize and define Sand as to define Pebbles or Rocks, the process is wrong. Add up the time you debate the merits, craft user stories, fill out the fields in JIRA, vote and prioritize into this sprint then re-prioritize into the next and the next, and assign and balance work across people. Don't forget to multiply meeting time by the number of people in the meeting. This can easily occupy more time than it takes to complete the item in the first place.

Sand will often originate from engineering; notice how many of the examples above are internal requirements for great software development rather than customer-driven requests. In these cases, it's often a waste of time to perform the usual formalities like user-story-writing, because the engineers already know what to do, and why. You can observe this waste in senseless force-feeding of uninformative "proper product language" and template-filling, e.g.

Type:	Task
Priority:	Moderate
Source:	Engineering
Estimate:	1h
Must-do by:	2022-03-15
Story:	As a software developer I want to upgrade npm package `minimist` from version `1.2.2` to `1.2.3` so that I don't have a security vulnerability.
Linked Goal:	Adhere to corporate security policies of applying security patches within thirty days of a known vulnerability of "low" severity and "low" risk.
Risks:	Might break unit tests, might require refactoring around new APIs or behaviors, might delay other work scheduled in the sprint

Testing Considerations:	Normal unit tests
Acceptance Criteria:	Nothing changes after the upgrade

It can get much worse, all for something that could just be a one-liner to "upgrade `minimist` to at least `1.2.3` because of a security patch," and in practice the work will usually be done in a few minutes, changing one line in `package.json` and re-running the unit-tests.

Prioritize with intuition and desire, not math and metrics

Because Sand is largely not measurable, complex prioritization systems won't produce meaningful results. Fortunately, exactly because they don't take long to implement, it's typically not important when they're done, and indeed many never will get done because we always have more ideas than time.

Therefore, this is a great opportunity to engage people's emotions, and go with things people *want* to do. Since delivered-value is near-nil, why not choose things people have energy for? This increases happiness, morale, and often quality. Because people naturally work harder and better on things they want to do, you get "productivity for free," which in turn increases throughput, which is the primary goal.

Self-managed teams schedule their own Sand

In keeping with the rule to minimize administrative overhead, teams should schedule Sand themselves, not debate prioritization with executives.

If the management is constantly engaged in Sand-scheduling, something is amiss and needs to be corrected, and it's always the manager's fault. Perhaps the team is perfectly capable of scheduling Sand; in that case, the manager is micro-managing, which is the manager's fault. Perhaps the team really doesn't understand the customer, the market, the technology, or the work, and therefore truly is incapable

of scheduling Sand; in that case, the manager has hired incorrectly or not created an environment where the team can understand these things, which is the manager's fault.

Little requests and suggestions are normal; indeed, these will come from all over the organization. But if you're debating with spreadsheets and virtual-sticky-note-boards, it's gone too far.

PEBBLES MAXIMIZE ROI

Pebbles are not a "balance" (p. 589) between Rocks and Sand; they are their own creatures. Their time frame is definitionally constrained between that of Sand and Rocks; the more important difference is that while Rocks are strategic, with a view towards winning over the next few years, Pebbles are tactical wins that have an impact in the next few months, attacking the challenges you're facing right now, or a great feature idea you can surprise customers with sooner than they expect.

Duration: 1-4 sprints

Pebbles take multiple stories and possibly sprints. Unlike Sand, they do need to have a measurable impact; you can't spend a month or two of the team's time and have nothing objective to show for it.

It is difficult to craft great Pebbles, because impactful things have a tendency to explode in effort. Anything longer than four sprints invokes the Hofstadter problem of time and impact, and therefore must be analyzed and prioritized as a Rock. If a Pebble starts expanding, you have two choices:

credit[246]

"We'll start with some small pebbles."

1. Reduce the scope of the idea so it can be achieved in a smaller time frame, or
2. Move the item into the "Rocks" category, where it will be prioritized appropriately.

In both cases, you often discover that the impact is no longer big enough.* Either this means the team needs to get more creative in how to deliver more impact with less effort, or maybe this idea simply isn't a good-enough use of your time.

Maximize ROI

If Rocks maximize strategic impact over the long-term, Pebbles maximize immediate impact in the short-term. Said another way: They are the "most effective use of the team's precious time."

* For (1), reducing scope might also reduce impact, in number of customers affected or in the magnitude of the effect. For (2) the impact might have been great for a one-month project, but too small when compared to other large projects.

Pebbles maximize ROI: A measure of value, divided by a measure of effort, resulting in a metric of efficiency. Modest value won in a short time, or more value over more time, are both great uses of time. What you cannot do is deliver little value but still take a long time.

Beware the surprisingly high impact of estimation error on ROI

The Hofstadter problem is magnified with ROI calculations, so you have to be especially careful, especially with classic frameworks like rubrics.

For example, consider a task that ends up producing 20% less impact and ended up taking 50% more time than expected—a common real-world result:

	Estimated	**Actual**
Impact	60	48
Effort	4	6
ROI:	**15**	**8**

This item has *half* the ROI than we originally thought. This is an immense magnitude of error, swamping the signal you thought you computed. Many other items' ROI will fall within this range of error, telling us that the noise from the error exceeds the signal from analysis. Which means the rubric is useless.

To avoid this issue, use this framework for performing an ROI analysis **(p. 171)****.**

Product Managers decide Pebbles

You could argue that the Product Manager's most important job (p. 817) is to make decisions exactly like this one: Which Pebble will we tackle next?

Ultimately it's a judgment call: A synthesis of what customers need most, what competitors are doing, what is consistent with the strategy,

what is best for company metrics, and what is best for customer delight. Input from many directions is appreciated, but one mind needs to make the call. That should be the mind closest to the customers, to the product, to the competitors, and to the market, not a drive-by decision from a manager, and not a hostage negotiation with engineers who would rather rewrite a whole module from scratch.

That said, committing a few months of time to anything is a big decision, which means it should have a sensible justification, and some objective measure of impact so we can see whether this activity is having the desired effect. With "self-managed teams" comes not only the freedom to decide and act, but the responsibility to own the results.

A SIMPLE SPRINT-PLANNING SYSTEM

How can you put all this together in practice, in real sprint-planning?

Schedule things in this order, skipping one if there is insufficient capacity to make significant progress on it given the other items in the list, or if high-quality stories aren't ready-to-work:

1. Time-critical items, regardless of size. *(Examples: security patches, bugs actively impacting customers, critical work for a launch or other event with an externally-imposed, immovable date)*
2. One or more stories from the current Rock.
3. One or more stories from the current Pebble.
4. Sand.

Life is never as simple as that, so here's how to manage the common issues:

"Time-Critical" takes up so much time, we can't make progress on Rocks and Pebbles
You have a meta-problem: Your team doesn't have enough time to be effective; solving *this* is now your top priority. This problem is even more important than your Rock, because in this condition you won't actually complete any Rock. There are myriad causes, and maybe multiple simultaneously: Is it a problem of individual productivity, of the team owning too many things, of architectural dependencies, of the problem-domain requiring more people, of lacking specialized skill sets, of greater fortitude of saying "no" to certain requests, or what? You must diagnose and cure the disease. Schedule sprint time to work on the solution.

Starving the Rock
Just one story per sprint will cause too much context-switching, and take too much calendar-time. If you're constantly starving your most strategic item, this is an impediment that the PM needs to address with the team. You're probably falling prey to the Eisenhower Matrix[247] fallacy of working on things that are urgent, rather than things that are important. Maybe you need to pause your Pebble for a while?

More than one Pebble
If you're truly going to execute on all four sections, you don't have time for two Pebbles at once. Plus, the context-switching is worse for morale and for productivity. The exception is when a Pebble is essentially complete, or will be blocked for at least a week, and therefore you truly do have the time to work on something else.

Always the Rock, never the Pebble
Because Sand is definitionally small, a few of those items always fit. But, a significant story for a Pebble may not. What if Pebble stories *never* fit? Consider that this might not actually be a problem, if the Rock is so valuable. Declaring the Rock "the theme of the next four months" might be exactly what the team needs for focus and maximum impact. It could be that once the Rock is "done" (in the sense of "first complete version"), you can then tackle a Pebble and then another, just doing incremental (small) updates to the Rock as you continue to learn and evolve its result, rather than immediately tackling an entirely new Rock.

Zero Sand
Since it's last on the list, and often aren't even full stories, it's easy to just never do Sand, but this will result in a poor product and unhappy engineers (as much of their internal work falls under this category). Consider toning down stories from other areas, maybe pausing the Rock or Pebble for one sprint, or even having a sprint devoted only to Sand, as a fun way to get a ton accomplished in a short amount of time and to break up the monotony of the sprint-planning cycle.

Starting a new Rock or Pebble too soon, rather than creating more quality and value from the ones that just "completed"
With long lists of genuinely terrific Rocks and Pebbles, it's tempting to start a new one as soon as the current one is complete. But software rarely works that way. Between learning how customers actually use (or don't use) things in the field, completing small items that were originally deferred (so that we shipped sooner and started learning sooner and started selling the feature sooner), and both incremental and significant follow-on functionality, often you need to keep the Rock and Pebble around longer than it first seemed, or at least not start a new one quite yet. This is realistic for great software and a healthy, sustainable pace of work, and this is another reason why our "time estimates" on both Rocks and Pebbles are subject to Hofstadter's Law, and further justifies our draconian admonitions about identifying and prioritizing that work.

Trying to "balance" every sprint
It's not important that every sprint is perfectly balanced between all types of work. It *is* important that we're balanced over a period several months, otherwise something important is getting starved. Indeed, it's often wise to build imbalanced sprints intentionally, because that means greater focus, less context-switching, and therefore getting more quality work done.

Of course all this is easy to say, but hard to execute. Still, when we write it down as simply as possible, and try to honor it, we'll make better decisions sprint by sprint, which in turn creates the most impact year by year.

Chapter 12:
The Iterative-Hypothesis customer development method

TRUTH · THE PROCESS · MAXIMIZING RESULTS

"I also managed dozens of employees and oversaw international travel arrangements. Plundering was actually a very small part of the job."

credit[248]

This is a simple but effective process for building knowledge through interviewing (potential) customers that I've employed multiple times in the past 16 years.

This system led to rejecting some startup ideas (p. 845), selecting the idea of WP Engine, and selecting the right features during the early years, which then led to hyper-growth, which led to a Unicorn company with three dozen teams who do customer development for themselves.

This article explains how to build and execute interviews. If you don't have anyone to interview, this article explains how to find potential customers to interview (p. 683).

> *"Deep in the forest there's an unexpected clearing that can be reached only by someone who has lost his way."*
>
> *–Tomas Tranströmer*

THE GOAL IS TO UNCOVER THE TRUTH, NOT TO SELL

I've been the interviewee for many startups doing customer development, and their most common mistake is that they spend most of the time selling me on how great their idea is.

If you—a reasonably intelligent, excited, passionate person—sit down with someone who meets the criteria of a potential customer, and make an hour-long sales pitch explaining all the features and benefits, with that person wanting to be kind and supportive of this

interviewer who is as passionate as they are desperate, that person will probably say something like, "yeah, that sounds pretty good."

So, what have you accomplished? Nothing. **If you don't come away knowing something new and actionable at the end of the interview, you've wasted your time and theirs.**

In the Lean Startup method they call this[249] "validating your ideas," so it's tempting to spend the time convincing the person to provide you with validation. Instead, your mindset should be: "What does this person know, that invalidates something I thought was true?"

If you've set out to confirm your ideas, not to disconfirm, then you will easily see the confirmation and conveniently miss the disconfirmation, and you will have done worse than waste your time—you have convinced yourself to believe incorrect assumptions.

This process is a specific way to achieve that outcome: Maximizing genuine learning.

> *"Listening is being able to be changed by the other person."*
>
> *–Alan Alda*

THE PROCESS

1. Goals: What you're trying to learn

What is it that you're trying to learn? If it's a new product at a new company, in a B2B space, where the fundamental value-proposition is to solve some specific existing customer pain-point, the list should include everything standing between the customer's problem and your potential product (p. 71); a list might be something like:

1. What does the "perfect customer (p. 317)" (PC) look like?
2. What outcomes does PC need to deliver in a typical month?
   (e.g. JTBD framework[250])
3. What does PC actually do in a typical day?
   (e.g. tools, workflows, things they love, things they dread)
4. What pain points does PC experience today?
   (i.e. what actually happens, and what pain does the customer actually know about?)
5. How does PC cope with that pain today?
   (i.e. what is your competition, including DIY?)
6. How much would PC pay to eliminate that pain? How is PC able to budget and execute the payment?
   (i.e. what are viable prices and terms?)
7. What is the triggering moment? What causes PC to decide: Today's the day I'm going to buy something?
   (because no one randomly switches vendors)
8. What causes PC to resist or fear buying?
   (habits of the present, anxiety of change, risk or cost of implementation)
9. Where does PC go to discover and buy products like this?
   (i.e. what are the best distribution channels?)
10. What specific words does PC use to talk about the space; what tacit assumptions does PC have?
    (i.e. how should you talk about the product?)

11. What ultimate, higher-level goal (p. 259) does PC have?
(i.e. what outcomes are they expecting as a byproduct?)

Decide on your list of goals first, as they will drive the content of your interviews.

Notice what is *not* here: Asking the customer what you should build, or whether would buy some specific feature.

This has happened to me dozens of times in my 25-year career: I asked a customer "Would you buy if we build _____?" and then say "Yes", and then we build it, and then they don't buy. Every seasoned Product Manager will regale you with the same story. This you cannot just ask that.

Here's where I'm supposed to trot out the Henry Ford quote: "If I asked my customers what they wanted, they would have said 'a faster horse.'" Except sometimes "a faster horse" is a wonderful product and a successful startup. And sometimes you should indeed invent a car. In neither case can you find out by asking customers what to build.

Instead, what you *can* learn from talking to customers, is what their current life is like, which answers the many important questions listed above, leading someday to Product/Market Fit (p. 9).

2. Hypotheses: Your current answers

It sounds funny to write down the answers ahead of time; after all, the whole point of interviewing is to empirically discover the answers, not to presume you already have them! "Learning" and all that.

The first reason to do this comes from the literature on the science of predictions. People are more objective at seeking the truth when they're forced to record their predictions—for example as formal "bets"—and observe how reality confirms or clashes with those bets. We autonomically retcon[251] our beliefs in the presence of new informa-

tion, or just discard the new information (i.e. Confirmation Bias[252]). Writing down our predictions helps us avoid this fallacy.

The second reason to write down hypotheses is that, as we're about to see, they will help us generate great interview questions.

Create hypotheses for each of your goals. At least one per goal, but more is fine. It's also fine to have hypotheses that aren't attached to a goal, if you're really curious about it.

Here's an example list of hypotheses I had about WordPress hosting in 2009 before I started WP Engine; I've added a "mapping" to the goal-list above:

1. [G1,G4] Bloggers with more than 100,000 page-views per month have trouble keeping their blog fast.
2. [G1,G4] Bloggers with more than 10,000 RSS subscribers have traffic bursts that take down their site, even if the site functions just fine under normal conditions.
3. [G4] All WordPress bloggers worry about getting hacked, because it's common knowledge that blogs get hacked constantly.
4. [G3] Serious bloggers spend at least 3 hours per day inside WordPress—whether writing or answering comments.
5. [G3] Serious bloggers spend at least 2 hours per week on IT tasks related to hosting.
6. [G5,G6] Some bloggers spend thousands of dollars on consultants to make blogs fast and scalable.
7. [G5] Some bloggers just live with the problems.
8. [G1,G6] A blogger with 50,000 page-views per month will pay $50/mo if these named problems go away.
9. [G6] Bloggers use personal credit cards to buy supporting software for their blog.
10. [G6] Bloggers need to try software before they're comfortable buying.
11. [G7] When bloggers get hacked, it's a traumatic moment in which they say "I never want that to happen again," and they're ready to switch hosting providers.

12. [G8] Scary, unclear, expensive to move all your data; what if you get there and it doesn't work?
13. [G9] Bloggers read blogs-about-blogging for tips.
14. [G9] Bloggers trust the advice of WordPress consultants.
15. [G2] Bloggers care about driving RSS subscribers more than anything else, because those are repeat viewers.
16. [G2] Serious bloggers publish at least four times per week, to stoke page views for advertisements, reader interest, and Google search results.
17. [G10] Bloggers call themselves "bloggers," not writers, authors, content-marketers, etc..
18. [G10] Bloggers call their website a "blog," not a "website."

Even if you don't know anything about "blogging in 2009," I'll bet this set of theories sounds reasonable to you. So reasonable, that maybe you'd agree it's not worth spending time validating them.

But you'd be wrong. (And so was I. We're all wrong, at the beginning.) After dozens of hours of interviews, I found that half of these hypotheses were wrong. The ones that were correct still needed to be tuned in detail. And equally valuably, I discovered additional attitudes and behaviors that weren't in my original list.

> *"The greatest enemy of knowledge is not ignorance. It's the illusion of knowledge."*
>
> *—Stephen Hawking*

For example: It's not true that most people are willing to pay extra for extra security. It turns out that selling security to bloggers is like selling backup software: If you've never had a hard drive failure, it's unlikely you'll pay $30/mo for a backup service. Once you experience

that devastating event, the first thing you do with your new laptop is sign up for any service that promises 100% full automatic backup. (Selling security to mid-sized companies is different; by then, they're proactively executing a formal security policy.)

Another example: While it is true that bloggers want to "try before you buy" for most software (whether "trial" or "freemium"), this was not the case with hosting their website. It's such a disruption and technical ordeal to move their website to a new vendor, they think of it as permanent, not temporary. Therefore, "free trial" is not the most compelling offer, whereas "free migrations" is. By the way, we tried "free trial" anyway—how can people not love a free trial!—but the conversion rate was over 90%, and when we removed the free trial, sign-ups didn't decrease at all.

It's important to list even the most obvious, mundane assumptions, because you'll be surprised how often you're wrong or they need adjustment, even if (p. 1427) you're an expert in the field.

Put these hypotheses in the first column of a spreadsheet, one per row.

3. Questions: What you ask during the interview

Generating good questions is the hard part for most people. Armed with your hypotheses, however, it becomes easy, if you adhere to the following system.

> *"Questions are places in your mind where answers fit. If you haven't asked the question, the answer has nowhere to go. It hits your mind and bounces right off. You have to want to know."*
>
> *—Clayton Christensen*

Questions are designed to test your hypotheses, and to suggest better ones. To achieve this, go to the second column in your spreadsheet, and write one question per hypothesis. Sometimes one question can cover a few hypotheses, if they're closely related.

Questions must be open-ended. This is where most people go wrong. They'll have a hypothesis like the "security" example above, so they'll ask a question that "leads the witness," because they're still in "selling" mode instead of "discovery" mode:

> Blogs get hacked all the time, and when they do it's devastating, right? Would you like it if your hosting company had extra security measures to protect your blog?

Of course everyone will say "yes." They would sound dumb if they didn't agree. That's why it's a useless question. You didn't find out what the person actually thinks. And therefore you didn't find out what everyone else will think when they look at your advertisement or arrive on your home page or review your pricing page. The assumption baked into your question could be wrong, and now you'll never know.

Instead, write open-ended questions that:

1. Confirm or negate the hypothesis *(the point of the exercise)*
2. Do not hint at any one specific answer *(seek unbiased truth)*

3. Invite the generation a specific answer *(uncover the correct answer)*
4. Invite more information *(seek answers to questions you didn't know to ask)*

On "security," for example:

> Do you ever think about website security? If so, how do you think about that? Do you do anything about it today?

Here's more examples from the hypotheses above:

Hypothesis	Bad Question	Good Question
Bloggers call themselves "bloggers."	Do you consider yourself a "blogger?"	When you meet someone new, how do you explain what you do in a few sentences?
Serious bloggers publish at least four times per week.	Do you publish often, so Google ranks you high in SEO and there's a lot of surface area for people to find you?	How often do you publish new content? Why at that rate—what led you to that decision?
Some bloggers spend thousands of dollars on consultants to make blogs fast and scalable	Would you spend $2000 on a consultant, if it meant your blog would become much faster and more scalable, so you rank higher on Google search results and get more traffic?	How valuable is the speed of your website? Have you ever spent money to improve it? If so, how much, and what did you do? Did it work? Were you happy with that investment?

4. Iterate the hypotheses

During the interview, take notes in the spreadsheet, in a new column, next to each question (which in turn is next to each hypothesis). The

conversion might deviate from the original point; that's OK, maybe straying will lead to learning new things.

> *"The most exciting phrase in science isn't "eureka," but rather, "that's funny."*
>
> *–Isaac Asimov*

If you hear anything surprising, ask follow-up questions. **Surprise means you're learning**, and since "learning" is the whole point of the exercise, you should use "surprise" as a signal that you should dig deeper. You can use an old interviewing trick: Just say: "Tell me more about that."

After each interview, consider what supported or contradicted your hypotheses. Should you alter some of them? Not necessarily, especially after just a few interviews, but definitely if you're seeing a pattern.

Also create new hypotheses (and associated questions) based on new learnings. The more insight you can get about your potential customers, the better. Don't worry about whether every hypothesis maps cleanly onto one of your goals; just accumulate insight.

> *"There are two possible outcomes: if the result confirms the hypothesis, then you've made a measurement. If the result is contrary to the hypothesis, then you've made a discovery"*
>
> *–Enrico Fermi*

5. Stop when it's boring

Do you have all the right answers? Who knows. Probably not. But when this method stops producing new information, then you need a new method for making progress. That could be a proof-of-concept, a high-fidelity demo, an MVP (or actually an SLC (p. 101)), or something else.

When the surprises stop, that means learning has stopped, and that means you should stop the process.

A typical mistake is to do three interviews, and then stop because "I'm not learning anything." With so little input, you might not be genuinely seeking to learn. That's like a marketer saying "I tried two variants of my AdWords ad, and none of them are better than my first attempt, so I'm not going to try any more variants."

It's also possible there's nothing to learn because you're fishing in the wrong spot. Maybe there's no patterns because you haven't found a real pain-point that more than a few people have and are willing to pay for. That means you need different ideas.

credit[253]

"So, that thing you're here to talk about…
Talk about that a little."

Here's more detail on how to determine (p. 845) whether the result of customer interviews is telling you "don't pursue this idea."

MAXIMIZING YOUR RESULTS

So that's the whole process.

Here are more tips.

Emergent segmentation

You might notice that customers are segmented. Meaning, a certain type of customer tends towards one set of answers, while another has a different set. In the case of security, for example, when you talk to marketing departments at large companies, they do think about security, whereas when you talk to independent bloggers, they almost never do.

In this case, it's useful to make a note of the segments you think exist. First, write hypotheses and questions that you believe are the determining characteristics of the segments. You'll ask these at the top of the call. Then, keep separate spreadsheets of hypotheses, one per segment. If you're lucky, you'll end up with clarity on the types of customers, and what each are like. You might choose to target one, some, or all of these; regardless, understanding the landscape is invaluable.

Discuss price

This one is controversial; many intelligent people insist that you shouldn't discuss price in early customer interviews because it un-

necessarily conflates financial considerations with the discovery of customer attitudes, behavior, and pain-points.

But in my opinion the price-tag is an essential component of the interview, because I believe the price is inextricably linked to what the product means to that person, therefore how they think about it and how it affects their life. Price also determines the business model of the company (p. 515), so it should not be "figured out later."

In early interviews for WP Engine, near the end of the call I would float a price of $50/mo for a service that made their website faster, more scalable, more secure, and came with genuinely good customer service. The responses were immediate, emotional, and vehement. One group was shocked—*shocked*—that the price tag would be so high; they would *never* pay even *close* to that amount. Another group said they would only buy the service if it were much *more* expensive, because otherwise they know it couldn't possibly fulfill its promises. (Those groups turned out to be emergent segments; see above.)

Had I not discussed price, I would never have learned about the segmentation, or the expectations behind those segments, and thus how to price correctly, and for whom.

I had another startup concept before WP Engine. Insights stemming from pricing discussions[254] was one of the primary ways I was able to invalidate that idea, which in turn created the space for WP Engine, which is now a unicorn.

Pricing questions can be open-ended (i.e. "How much would you expect to pay for ..."), however I think quoting a specific price is an acceptable breach of protocol. Putting a specific price in front of people elicits a strong, visceral response. When someone actually visits your pricing page in future, this is also the experience they will have—

reacting to a specific price. It's smart to test what that experience will be like.

You can also tie pricing questions into your other questions, to test whether the person really values that topic. For example, another way to test the hypothesis that bloggers care about security would be:

> Would you pay extra for a security package that really worked, or do you not really worry about being singled out for getting attacked by a hacker?

By asking if they'd pay extra, and by almost suggesting that they shouldn't bother, you're testing whether they *really* ascribe value to the concept. This worked in practice—most bloggers initially claimed security was important to them, but admitted they wouldn't pay extra to have more of it.

Expect contradictions

You're going to get all sorts of contradictory signals. People are different. Sometimes because they have different goals, different values, different past experiences, different roles, different projects, or for no discernible reason whatsoever. So your data is going to be noisy.

Some of your hypotheses will end up reflecting the variation. You might conclude that some number varies a lot rather than staying in a small range, or that there is no pattern in people's opinions about some topic. That's still learning: Knowing what patterns *don't* exist prevents you from making false assumptions.

Real patterns will stand out from that noise; that's your fundamental truth, that you can build products and strategies around. There might not be much of it. All the more reason to highlight it.

Ask them to explain, step-by-step, how they will use it.

The pattern: You ask a customer if they have the problem; they do. You ask whether they'd buy your product to solve it; they say yes. Then you build it, and they don't buy. Somehow, your interrogation didn't work.

Sales conversions are never 100%, but one technique is to ask them to describe, in painstaking detail, exactly how they will use the product in their daily life. When would they open it up, how does it fit into their workflow, which features do they invoke, how do they move the outputs into other systems?

This works because while they really do mean "yes, I think that *sounds* good," thinking it through uncovers barriers that in fact blocks the sale; it turns out they need it to integrate better with something, or they actually need a feature you weren't contemplating, or some other hiccup.

Create your positioning from customers' exact words

Discord uses the oddly cold word "server" to mean "room" or "community." Why?

Founder Jason Citron discovered early on that kids were setting up virtual servers to host audio or chat. So Discord's early pitch was: "Get a free server!" While it might sound better in an investor deck to say "we create communities," that wasn't how to sell the product.

Use your customers' words, not your own. Discover those words, by recording the meetings and noting exactly what they say.

Everyone "knows" what everyone else wants, except they don't

I don't think I've ever conducted a customer interview where the interviewee didn't switch into "market guru" mode. This is where the customer stops talking about her own life, her own problems, what features or price tag would be acceptable to her, and starts talking *on behalf of other people*.

> "I wouldn't pay $50/mo, but a lot of people would."
> "I would pay $50/mo, but most people would expect this to be free."
> "I care about security, but most people are completely clueless about that."
> "Well I use a free tool to do that, but most people don't."

Of course it's coming from a good place—they want to help you, they want to explain the "state of the market," they want to leverage their expertise. But they don't know what everyone else wants.

Neither do you; that's why you're doing these interviews. And when you see all the crazy, different things people think, you realize that it takes tons of interviews to uncover even a modicum of truth. The person you're interviewing hasn't done that, so they don't know the truth.

One of the hallmarks of successful companies is that they found some untapped aspect of the market and owned it. That might be a feature no one had considered, or a technology that before now hadn't been accessible, or a realization that the human touch trumps everything (Zappos) or that the human touch doesn't matter at all (Geico). Even if the person you're interviewing is a market expert, you're specifically looking for interesting holes and niches that the experts haven't noticed!

So be polite, calmly tell them that's great insight, reinterpret "everyone else wants X" to either mean "I want X" or just throw out

the comment completely, and redirect the conversation back to themselves and their own specific situation.

Leveraging AI... maybe

AI might help. Perhaps it could...

- Generate hypothesis from the goals, adhering to the guidelines, specifying which goal attaches to which hypothesis *(although of course you'll treat them as templates and correct them to what you actually believe)*.
- Generate questions for the hypothesis, specifying which question attaches to which hypothesis, adhering to the guidelines above.
- Clean up transcripts.
- Scan conversations for common themes.
- Scan conversations for "things that contradict a specific set of hypothesis."
- Scan multiple conversations for themes.

However, **half the value of this exercise thinking through this stuff for yourself.** The "aha" moments come only when you wrestle with the details.

You find the contradictions. You discover maybe you didn't think that after all. You realize new ideas can solve for conflicting inputs (p. 589). You realize which hypothesis are right, wrong, different.

So, I advise you *not* to use AI, except to accelerate busywork such as:

- Clean up transcripts so they occupy less space and are easier to process.
- Double-check your thinking *after* you do the thinking, maybe come up with new ideas.
- Summarize your thinking, i.e. take your thinking that you wrote out or said aloud, and make the result pithy and clear.

STFU

If you're talking, you're not learning. To maximize learning, minimize talking.

When you do talk, it should be because you're clarifying your understanding about what they just said, digging deeper on the current conversation, or opening up a new vein of conversation.

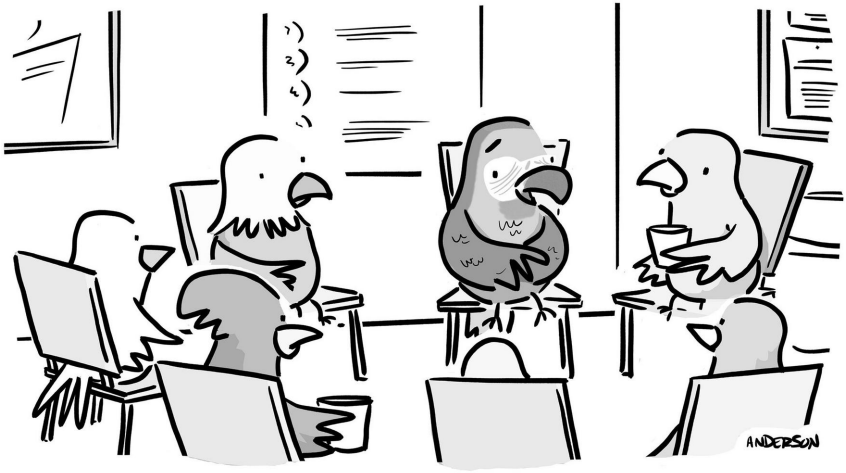
Further reading

- 40 Tips for B2B Customer Development Interviews[255] (by SK Murphy[256]), lots of specific tips, plus a list of even more articles on the topic.
- 12 tips for customer development interviews[257] (by Giff Constable,[258] 2012, just as relevant today as then, wholly compatible with this process.)
- 11 Customer Development Anti-Patterns[259] (also by Giff Constable,[260] 2013; sometimes it's easier to list what *not* to do.)
- Customer Interviews: Get Actionable Insights from Every Interview[261] (Comprehensive advice from Teresa Torres,[262] also the author of the fantastic book *Continuous Discovery Habits*[263] that also explains what to do with the information you get from interviews.)
- *The Mom Test*[264] by Rob Fitzpatrick (2013)—probably the single best book on this topic, and compatible with all the articles on this site.

credit265

"Thanks for coming in. You've given me a lot to forget about after you leave."

Chapter 13:
Using the Needs Stack for competitive strategy



"I knew it was bad when I stopped wanting a cracker and started needing one."

THE NEEDS STACK

How would you describe what is Charlie doing?

Charlie creates an account with AWS—Amazon Web Services: pay by-the-hour and by-the-gigabyte for internet servers, storage, and network connectivity—happy to see a "free tier" for the infrastructure they need, just like the "how-to" article explained. All the acronyms are new and unfamiliar; this is tricky, but exciting! Someday, Charlie's website might grow large enough to exceed the free tier—let's hope!

Apparently, what Charlie is doing is:

★ Buy infrastructure: web server, disk space, networking, database.

Nominally this is true, but Charlie's life ambition is not "to buy infrastructure." In fact, it's a means to and end. What Charlie *really* wants is to have a functioning WordPress-based website, because WordPress is by an order of magnitude the most popular* way for people to easily write and publish content, and integrate with online marketing software needed to build a successful website.

So, a better way to explain what's happening, is that Charlie has a higher-level "need," which is to set up a WordPress site. Getting the infrastructure is a means to an end, rather than the end itself; in this sense, it's a "need" but it's lower in the "stack."

What Charlie is doing *really* is:

---

* 43% of the largest 10,000,000 websites use WordPress; Shopify is the next largest with 4%. (Data from W3Techs)[267]

★ Set up a WordPress site.
→ Buy infrastructure: web server, disk space, networking, database.

This might sound like a verbose restatement of Jobs to be Done[268] or Five Why's,[269] but **there's a strategic competitive insight lurking here**, which companies often fail to appreciate, at their peril:

Charlie never wanted to set up infrastructure, and still doesn't. Charlie wants a WordPress site. Therefore, a company who provides the higher-level need of "WordPress site" makes the lower-level need of "infrastructure" *obsolete*.

Specifically, when a company like WP Engine[270] makes it possible to set up a WordPress site in one minute, with no infrastructure to think about, handling all the technology and 24/7 monitoring, and customer support that helps with WordPress, then that company is going to win Charlie's business, and AWS will never see Charlie.

Charlie disappears from the cloud infrastructure market. This is the insight.

If you ask AWS who their competitors are, the answers are other cloud infrastructure suppliers like GCP (Google), Azure (Microsoft), and Digital Ocean. And they are correct, because the Charlies of the world who *do* buy infrastructure, will buy from one of those. But WP Engine is also a competitor of AWS when it comes to Charlie, not because WP Engine competes on infrastructure (which it doesn't, and in fact WP Engine itself buys infrastructure from AWS, GCP, and Azure!), but because WP Engine targets a higher-level "need" in the stack.

But you can't stop there. That's the trouble—you always want to stop when you've reached the level you operate on, but there's always another level.

> *"Instagram's real product isn't photos; it's likes."*
>
> *–Alex Danco*

It's not Charlie's life ambition to make a WordPress site; this is just a means to an end. What Charlie *really* wants is to have a personal website for content and self-promotion. Charlie wants to carve out a digital *pied à terre*. In the virtual universe there's infinite *terre*, so why not slap your *pied* on some of it!

Now the Needs Stack becomes:

★ Have a personal website for content and self-promotion.
→ Set up a WordPress site.
→ Buy infrastructure: web server, disk space, networking, database.

Charlie can still achieve this higher-level need with WordPress running on WP Engine—indeed, over one million websites run this way today. But there are alternatives that target this higher-level need. Wix and Squarespace both let you create websites without WordPress, cutting out the step of "get WordPress" just as WP Engine cut out the step of "get infrastructure."

Does this mean WP Engine no longer has a business? No, that's not the conclusion, any more than you should conclude that AWS no longer has a business. WP Engine is 15 years old and still healthy, growing, and profitable; AWS just posted $5.2B in *profit* just last *quarter*. So what's going on?

These alternatives to WordPress certainly have advantages, but they are worse than WordPress for long-form content and for customizability. For this reason, they might not be right for Charlie. But they *are* competitors. Similarly, WP Engine doesn't let you custom-

ize your infrastructure, so those who need to do so, still need to buy directly from AWS.

The pattern here is the next insight: **Up the stack, customers achieve their end-goal faster; down the stack, customers have more flexibility and customization.**

But wait... once again "having a personal website" is not Charlie's life ambition, but rather just a means to an end. Charlie *really* wants a book deal. Being a properly-published author with a hard-back volume on the shelf that you can show to your kids[271] (who will just shrug today but maybe someday they'll appreciate it), and that you can gaze upon now and then in your dotage as a great accomplishment. This is Charlie's personal idea of "successful and popular." The way to get a book deal is to already have an online following; you'll have honed your writing craft, you'll have a natural audience to kick-start sales, and the fact that you've won followers in the competitive, noisy Internet proves your content is worth reading. Publishers want authors whose content has been de-risked and who come with a built-in marketing channel.

So the Needs Stack deepens:

★ Become a popular content-producer, leading to a book deal.
→ Have a personal website for content and self-promotion.
→ Set up a WordPress site.
→ Buy infrastructure: web server, disk space, networking, database.

The new layer brings new alternative solutions. There are companies like Substack, purpose-built to help you gain a following, with the tools for long-form content, for building a mailing list, and for promoting your content on social media.

Substack doesn't automatically win, because WordPress is better than Substack in many ways. For example, WordPress is far more configurable, so you can make it look unique, unlike a million news-

letters at Substack that all look identical (with character-kerning that's too wide for my taste, but I'm biased). Also, the tools for long-form writing in WordPress are far superior, with drag-and-drop elements supporting complex layouts—including different layouts for phones versus tablets versus laptops—that an email newsletter cannot support. If Charlie values those things, then WordPress is still the right choice.

This again exemplifies the pattern that software at higher levels of the Needs Stack have fewer features and less customization than software lower on the Stack. In exchange for these limitations, their focus on the higher-level Need means it's easier for the customer to achieve that Need. If the ease of achieving the Need is more important to a customer than the ability to customize and extend, the higher-level product makes all lower levels obsolete.

But wait a minute Substack because—you guessed it—there's another level to Charlie's Needs Stack. Because while Charlie wants a book deal as a signifier of popularity and success, it's not Charlie's life ambition. Charlie *really* wants to become a famous speaker. It started at a Tony Robbins[272] show; Charlie instantly knew that the stage was their calling. It's the prestige of being "that person" at parties, and the idea of being a jet-setter in first-class and a black car to pick you up with your name in the window, the ego basking in the glow of the lights and the elevation of the stage, gazed upon by an adoring audience who rushes up afterwards to grab a selfie and nervously blurt out a fraction of a story about how the speaker changed their life.

And the money.

★ Get lucrative speaking gigs around the world.
→ Become a popular content-producer, leading to a book deal.
→ Have a personal website for content and self-promotion.
→ Set up a WordPress site.
→ Buy infrastructure: web server, disk space, networking, database.

Charlie's plan isn't bad, but what if it were possible to skip the website and years of writing and self-promotion and the book deal and the two years of labor creating the book and directly become the next Tony Robbins? It's not impossible; there are speaker bureaus and online services like SpeakerMatch[273] that help you find gigs and facilitate the transaction. Why spend years building an online following (if you're lucky) when you could spend that time creating inspirational presentations, delivering them to acclaim, earning testimonials for your next gig, building a reputation with an agency, and getting bigger and better gigs over time. In this scenario, Substack, Squarespace, WP Engine, and AWS are all irrelevant for Charlie; with the higher need met, the rest might as well not exist.

The moral of the story is: You need to understand the Needs Stack of your target customers. Not just your "market" or your "product" or "solving the problem (p. 71)" or even the immediate Jobs-To-Be-Done of your customer—those are all means to ends.

Already you can see how this will help you think about positioning and features; let's see what else it helps you to do.

HOW TO LEVERAGE THE NEEDS STACK

The solution to "selling benefits" vs "selling features"

"Sell benefits, not features" marketers have told us since the universe cooled enough for galaxies to form. We're supposed to say "saves time" rather than showing a screenshot of the feature. (Or better yet, "grows revenue (p. 165).") For Charlie, say "Become famous" rather than "Get a website."

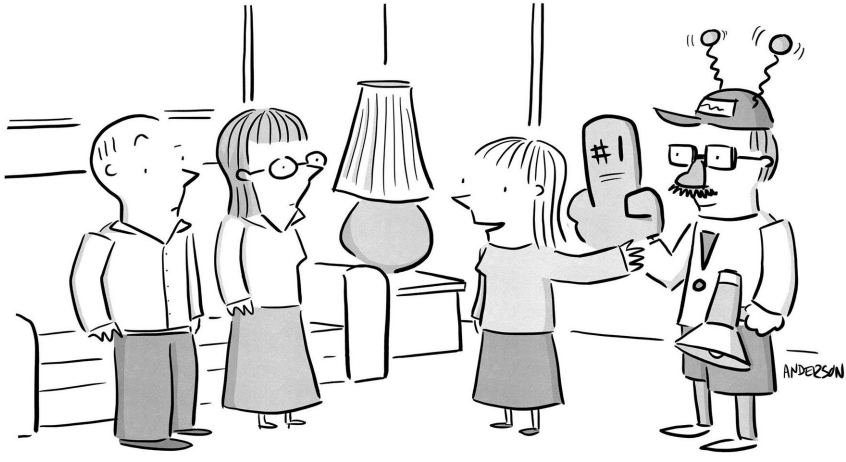
And yet, does an Engineer looking for a database infrastructure solution *really* want to be told "saves time" or do they want to see specs and features? (A: Specs and features.)

If a construction worker is looking for a wrench, do they search Google for "wrench that saves me time" or "socket wrench for a 5/16" lock nut?" And when that person clicks a link to potentially buy the wrench, do they want pictures of happy people "saving time," or do they want to see the specifications and features like "has a mounted light." Do they really need to be told "the light is useful to see in dark corners?" Oh! Thanks for spelling out the benefits of photons!

So which is it?

The Needs Stack suggests an answer. Charlie is not constantly and consciously thinking "I gotta become Tony Robbins," even though that's the top of their Needs Stack. Charlie is thinking at some other level of the stack at any given moment, for example "I gotta get some content online."

Google searches are a good proxy for "what level of the stack are potential customers operating at right now?" When Charlie embarked on this project, the search might have been something like "best way



credit[274]

"This is Tom. He creates awareness."

to publish online," not "how to become a famous globe-trotting speaker." After reading some articles about how WordPress is inexpensive, popular and has all the tools you need, maybe then Charlie moved down the Stack to search "how to get a WordPress site."

If someone has in their head "get a WordPress site," you want to meet them where they are, and talk about how your product is the best way to "get a WordPress site." **That's the part where you talk about features.**

However, benefits still have a role to play, because it's true that these features are a "means to an end," and the more you convince the customer that you're the best means to that end, you win even over alternatives. So, you can **sell the benefits that are one level higher on the stack.**

In this case, you could imagine a home page message like: "Become famous online with our turn-key, customizable WordPress website." That's the benefit from one level above, promised as a consequence of the current level.

Conversely, if you're Substack, you can't say "customizable" because that's not what the product is, and it's barely even a "website" (it's just back-issues of your newsletter), but you *can* speak to the product "getting eyeballs." So, that message could be "Grow your own audience with our turn-key newsletter platform."*

Position lower levels as irrelevant

If Substack tried to fight the battle at the "get a website" level of the stack, they would lose every time—it has almost no features of

* Sure enough, as of this writing the Substack homepage says: "Substack lets independent writers and podcasters publish directly to their audience and get paid through subscriptions," which are benefits from one level higher (i.e. "be independent" and "get paid for writing"). Lower on the page it lists features from the level they are on: "A Substack combines a blog, newsletter, payment system, and customer support team—all integrated seamlessly with a simple interface. We handle the admin, billing, and tech."

a modern website. But, it can fight the battle at the "build an audience" level; for some subset of the market, that's a more relevant level. Therefore, the idea is to make all levels below that irrelevant, by focusing only on your benefits:

"While others just give you a website, we help you build a following." Right, because "popularity" is what I wanted, not "a website!" This message won't work on businesses, for example, because they actually do need a website. But for the aspirational speaker, this might be perfect.

Conversely, WP Engine does the same thing with cloud services. The major cloud services say: "Oh it's easy to have a WordPress site. You just click 17 times on all this stuff to make a server, and then download and install a bunch of stuff to create server software, then set up and configure WordPress, then use this 8-page tutorial to integrate with a fast caching network (which costs extra), then download and configure WordPress plugins to talk to the caching network, then stay on top of all the security patches and software upgrades, and it wouldn't hurt you to learn how to SSH into your Ubuntu server so that...."

Whereas, the Managed WordPress Platform says: "You don't want infrastructure, you want a WordPress site, and in less than one minute, you'll have exactly that. And it's faster and more secure than doing it yourself. And when you have trouble, you can call our support line, which cloud services don't have."

In short, make "infrastructure" irrelevant by focusing on the next level up. The customer never wanted to think about infrastructure in the first place.

Add more value by moving up the hierarchy

If products targeted at one level disrupt the products below them, there's an obvious strategic conclusion: You could **disrupt yourself and your competitors** by moving up the hierarchy.

"Disrupting the marketplace sounds like a lot of work. Couldn't we just make a big mess instead?"

credit[275]

There are huge barriers to accomplishing this. The barriers are so large, companies almost never overcome them:

The next level is a different product, and a different business.
If you're known for selling WordPress sites, you'd have to re-brand to be known for building an audience, whether that's actually a new brand or expanding an existing brand. You also need to become expert in what that product is, and build it.* You might need to change your business model, e.g. selling infrastructure by the gigabyte, versus selling whole websites by the month, versus selling new subscribers on a pay-for-performance basis, versus taking a percentage of speaking fees.

The next level isn't possible
It's not clear that one *can* build a product that fulfills the promise to get someone a substantial following. Twitter and Facebook sort of can, in that you can pay them to advertise your content. But the Internet is a large and noisy place; marketing software doesn't automatically generate

* Or buy it, which mature companies often do, trading cash for speed-to-market.

attention. Once you get the attention, if your content isn't compelling, software can't force people to subscribe. In short, if you go high enough in the Needs Stack, you might run into something that a product cannot produce. (Or at least *you* can't.)

The next level targets a smaller market
Few people share Charlie's Needs Stack exactly. A business with a broad customer base like WP Engine sells an all-purpose platform for all kinds of sites. There are many different personas whose Needs Stack includes "set up a WordPress site."* In each case, the "next level up" is different. Therefore, targeting a higher level also means narrowing the market. This might be wise—it is a tried-and-true strategy to be the best product in a smaller niche, rather than a small fish in a large, crowded ocean. But, a company who has an established brand in the wider market might not want to narrow into a sub-market. There might not be enough money in it.

Since a full pivot is unlikely to succeed, there are other ways to build a strategy that partially climbs the hierarchy:

Create sub-brands / products
Keep the original business going, and launch "vertical" or "niche" products. Leverage the scale and operational excellence of the parent, but treat each product as its own "startup."

Add features without fully committing to the next level
Perhaps you can't completely pivot to a product that promises subscriber-growth, but you could create an add-on product that helps someone do that. This becomes an extension of an existing product, aimed at personas who share that next level of their Needs Stack.

Better marketing
Address the need with words instead of features. Without "promising" results, you can help, educate, and lead in that area. You can have a field guide about how to attract and retain subscribers to a blog. You can give

* e.g. individuals wanting attention, companies broadcasting their brand, media companies attracting eyeballs for advertising revenue, eCommerce stores using content-marketing to drive sales, communities using content to inform and engage, non-profits raising money, governments interacting with citizens

customers ideas for new kinds of content. You could partner with other companies who help with promotion. You could highlight customers who have been successful, wherein they share their experience to help others.

Still, while this advice is good for most organizations who cannot completely move up the hierarchy, there are examples of fantastic businesses that succeeded exactly because they successfully made the transition. For example, all of the case studies given in the "Emotional vs Functional" chapter of *Blue Ocean Strategy*[276], such as Cemex selling "build an addition to your house, for your family" instead of just selling cement or Starbucks selling a lifestyle experience rather than B-grade coffee.

Measure the next level up

The perennial question: Which few key metrics (p. 645) measure whether the customer is deriving value from our product, and thus predict whether they'll stay a customer? And beyond that, become a vocal advocate? The Needs Stack helps us uncover these more strategic metrics.

Metrics from the bottom of the Stack measure whether we're operating well, and certainly this is necessary even if not sufficient. Metrics at the level the customer is actively thinking about are useful to measure whether our Product is delivering on its direct promise, which again is vital.

It's strategic to measure further up the stack. Not too far, otherwise it's not only difficult to measure, but too many external factors drive the number. But the very next level up could be a critical way to measure value and to understand whether customers will be happy in the long run.

If a customer of Shopify successfully creates a store, the product of "have a store online" is succeeding, and that will be reflected in metrics like "number of products currently for sale" and in user surveys

like "how easy was it to set up products for sale." That's important, but what if no one buys their wares? That customer will probably churn, even though the software delivered on its promise. Indeed, Shopify has enormous cancellation rates that would sink most companies; they survive because the ones that stay, grow a lot. Because they have so many inbound new customers, they can manage around the fact that only 34% stick around for even one year.[277]

Can Shopify ensure that all customers who create a store end up doing thousands of dollars a month in business? No, this is an example where the next level up is vital, but Shopify cannot control it. However, they can measure it, and help. They could have articles about how to do online marketing, or features inside the store that help potential customers complete a checkout rather than abandon.*

This is why Shopify's primary success metric is GMV;** they report it to Wall Street[278] directly after revenue, and in the same large font. GMV measures whether their customers are succeeding in the next level up on the stack. GMV not only reflects the truth about where the customer value is, it also encourages their product mangers to think about how to stretch up from the level of the stack they fully control.

A source of higher purpose

Our direct goals are often simple: Sign up more customers, get them to activate on the product, reduce churn, reduce support tickets, reduce costs. All of that is important, but none of it creates a higher purpose (p. 1169).

What's it all for (p. 827)? Why should we even bother?

---

* Features like ShopPay, in which previous customers of other Shopify stores are fast-tracked through the purchase process, increasing the percentage of shopping carts that result in completed transactions.

** Gross Merchandise Value: the total amount of money their customers are selling through their stores.

Many companies can't answer the question. Indeed, maybe they have no purpose other than "growth." Will folks be motivated to do their best work at a company that values only conversion rates?

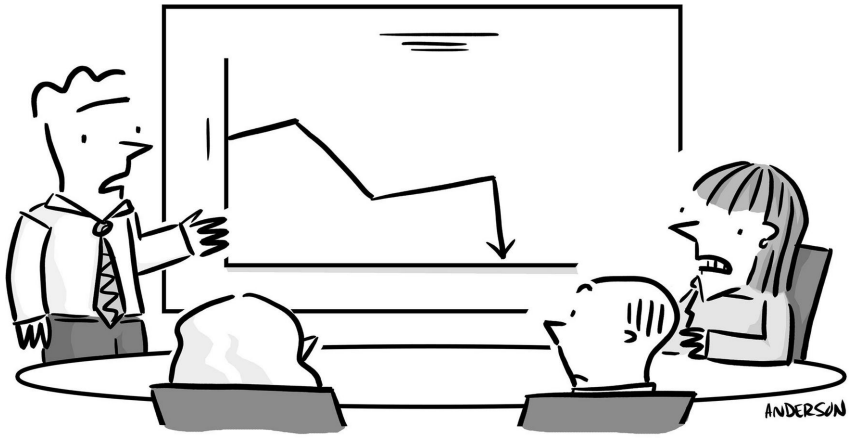
Purpose can be found in the higher levels of the Needs Stack. The immediate work is "getting a website," but the higher purpose is to enable someone to build a career for themselves, to engage their creativity and speciality, to help them win the respect of others. Or help them become a hero (p. 165) within a bigger organization through some new marketing project, which will eventually lead to their promotion. Or help them raise awareness and money for an important cause. Or enable them to speak truth to power in a region of the world where such speech is forbidden.

Tell those stories. On the website and internally. Celebrate with your team when this actually happens for a customer, because that *is* what it's all about. Those stories are why we should care.

Identify those upper reaches of the Needs Stack, not only because it helps you build more valuable features, not only because it helps you build a better strategy, but because it is the ultimate reason why any of this is worth doing in the first place.

Chapter 14:
Willingness-to-pay: Creating permanent competitive advantage for the right reasons

TRADITIONAL · "WILLINGNESS"
THREE KINDS OF WTP · LOVE · UTILITY · COERCION
EFFECTS · CREATE & SPLIT · NOT CHARITY
APPENDIX



"OK, let's not get into profit shaming here."

credit[279]

TRADITIONAL ECONOMICS: WTP AND CONSUMER SURPLUS

The best businesses deliver $4 of value, charge $2, and costs them $1 to do it.

It's an obvious formula for both profit and happy customers, but what does "$4 of value" even mean?

Economists have labels for this formula (Figure 1).

Willingness-to-pay (WTP) is the maximum price the customer *would have* paid for the product, which the economist claims is how much the customer *values* the product. "Value" could come from anything—utility, pleasure, status, even irrational confusion. The economist claims that any transaction is evidence that WTP > Price, and the difference between those numbers is "Consumer Surplus."

It looks trivial at a first glance, but I've come to believe that analyzing "WTP" is not only non-trivial, but also leads to very different strategies, business models, and outcomes.

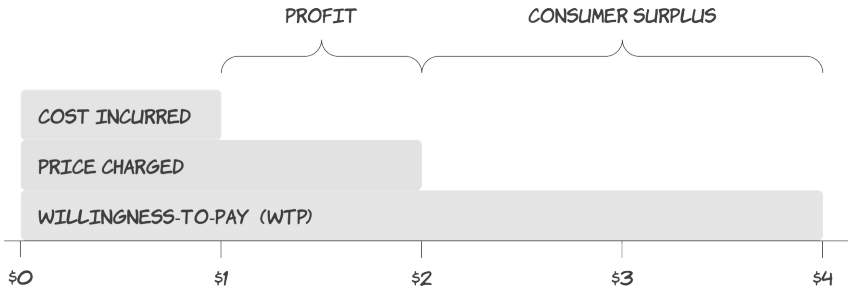


Figure 1

"WILLINGNESS" TO PAY

I'm irked by this word "willingness."

In 2015, Martin Shkreli, then-CEO of Turing Pharmaceuticals, bought the rights to the drug Daraprim, which for 62 years had been used to treat a deadly parasitic disease. He raised the price of a pill from $13 to $750, skyrocketing[280] the typical cost of treatment from $1,000 to $63,000.

"Profit" was his only justification for this abuse, in his own words:

> "I think it will be huge.... So 5,000 paying bottles at the new price is $375,000,000– almost all of it is profit, and I think we will get 3 years of that or more. Should be a very handsome investment for all of us." *–Martin Shkreli, in communication*[282] *with investors*

credit[281]

Martin Shkreli testifying before congress on a hearing on drug prices, before calling lawmakers "imbeciles"

Patients have no choice: It's pay or die. The economist would say, patients objectively have a high "willingness" to pay. But is this how we should define "willing?"

And when patients cannot afford a $63,000 treatment, and therefore don't purchase the drug, and die, should we say "well, I suppose they weren't 'willing' to pay?" This phrase captures neither the intent nor the ability to pay, both of which are critical factors in questions of price, profit, and consumer surplus.

While there are many* such examples, it's more instructive to point out mundane, non-life-threatening examples of why "willing" is not the right word.

It happens with commodities, which economists say are a "perfect market."[286] When crude oil prices go up, prices at the pump go up immediately, even though costs haven't yet risen. When crude oil prices go down, prices at the pump go down slowly, even after costs have in fact fallen. The same thing is happening now with eggs.[287] Is that because we're all "willing" to over-pay for gas and eggs?

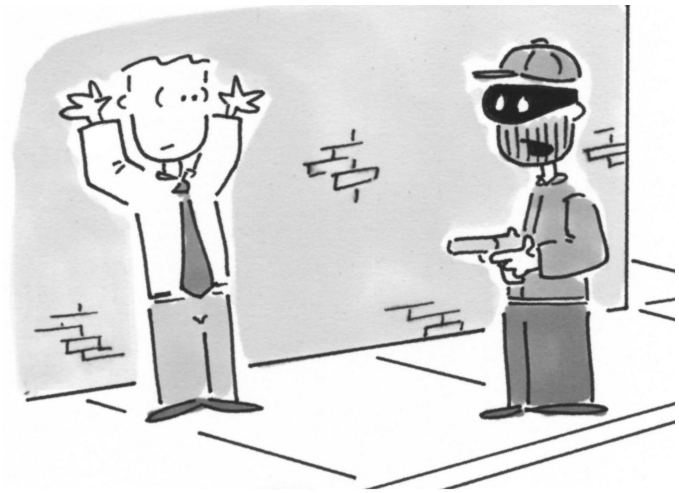
It happens with bundling—often touted as a wonderful strategy.[288] I never liked paying for cable TV, because it seemed expensive considering I still had to watch ads all the time. Most of the channels I paid for, I didn't watch. Cable companies know that of course; they bundle channels specifically because they know consumers are not "willing" to pay for all of them. Because the content-owners have a near-monopoly, consumers have no choice. Even with modern streaming services the problem persists, because whether it's Hulu Live or YouTube TV, it's still bundled, and still the same price.

There's also "willing" versus "able." Perhaps many more consumers would be "willing" to pay $1000 for a fully tricked-out smartphone, but most are not "able." This is vital fact when determining strategy, business models (p. 515), and company viability (p. 71), but an economist would just say "few consumers are 'willing' to pay $1000 for a high-end phone."

But it's not all bogus. There *is* a genuine concept of being "willing" to pay more, and thus genuine "Customer Surplus." I am willing to pay more for Anker[290] products (power strips and chargers) because they're extremely high-quality; I don't even notice if there's a competing product that is 20% cheaper. I'm loyal even though there is neither lock-in nor recurring revenue. People pay more for TOMS** and Patagonia***products because of their authentic missions. People

---

* There were at least four egregious cases[283] in 2015 alone. More recently, Moderna quadrupled the price of their COVID vaccines, its CEO Stephane Bancel saying[284] that the new price is "consistent with the value" of mRNA vaccines at 45 times the manufacturing cost, after the US government paid them billions[285] to cover the cost of developing the drug. Are we "willing" to pay even more? "Yes" in the sense that human life is valuable, but "no" in the usual sense of the word "willing."



credit[289]

"I prefer to call it profit sharing."

routinely pay more for coffee that has a fair and sustainable supply-chain, because they're willing to pay more to have a positive impact on the world, not just to consume the product.

** Blake Mycoskie was vacationing[291] in Argentina, when a knowledgable American opened his eyes to the outsized impact that a lack of shoes has on poor children. Unprotected feet are susceptible to punctures and infection, and prohibit walking long distances, which in turn means one cannot go to school. He founded TOMS shoes, selling an Argentinian-style shoe, with the logo of the Argentinian flag, with a marketing strategy he dubbed One for One: Every time you buy a pair of shoes, TOMS would give a pair to a needy child. After TOMS's financial success, Sketchers copied the strategy exactly, even down to the style of the shoe, the name ("BOBS"), and the altruism. Consumers were so outraged by this inauthentic strategy, Sketchers was forced to canceled the product line after just 24 hours (although they revived the brand later with a different mission). That strategy was individual to TOMS; it was irrelevant that the strategy was publicly visible and copy-able. TOMS has weaknesses—people complain[292] about poor customer service and shoes quickly developing holes—but they win anyway on the strength of the individualized story.

*** Besides their publicly-lauded sustainable practices and an outdoor-worshipping culture, they even have a formal company policy[293] to bail employees out of jail if arrested while protesting peacefully.

Indeed, genuine "willingness" creates the best, most durable, most profitable businesses. Consumers not only pay more, they're *happy* to pay more, creating profit margin. They become evangelists, driving efficient growth. The company is resilient to competition, because consumers are buying for reasons other than "features" and "price." The world becomes a better place, transcending a zero-sum game of winners and losers.

Analyzing the differences between these kinds of WTP yields insights that all products and companies can leverage (p. 543) to build the best strategies.

THREE KINDS OF WTP

I divide WTP into three categories, each having different drivers, and much different strategic value:

LOVE

- Mission: the joy of supporting a change that's bigger than all of us (p. 399), or a community or movement you want to see flourish
- Reciprocity: when the company gives before taking, or gives more than it takes, or provides exceptional customer service, or is deeply human.
- Exceptional design: a joy to use, a product that seems to genuinely care about your experience
- Exceptional quality: the pleasure and relief generated by reliability
- Personal identification: leveraging the company's brand as visible component of your own personal brand
- Culture: supporting an organization that treats employees and vendors well *
- Social or environmental impact: supporting sustainable, fair practices
- Community: a welcoming space where members learn and teach, support each other, create personal connections, grow their career or business, be part of a tribe
- Ecosystem: wherein all members make more money or gain more prestige than had they not been part of the group

* Counter-example: Walmart and Amazon, known for exploiting workers and suppliers

Result: Allyship. Consumers are genuinely happy to do business with you, and root for your success; when you make a profit, they cheer, because they want you to thrive; they advocate for you publicly, * *tying their personal brand with yours; they don't even consider the competition; the old saying that "people buy from the person they like;" they would be OK with a small price-increase.*

UTILITY

- Cheap: even if quality and functionality is low, it's better than not having the product
- Integrations: providing functionality while also more difficult to switch vendors
- System-of-record: being the official place for important data, making it risky and expensive to switch vendors
- Training: invested in having trained an organization, making it expensive and disruptive to switch vendors
- Market-share Leader: the social-proof of selecting the market-leader is a reason to buy
- Location: coffee inside the airport is more expensive than on the street corner
- Convenience: groceries delivered to your doorstep are substantially more expensive than getting them yourself
- Simplicity: surprising ease is as delightful as it is useful

* This single tweet[294] demonstrates this with thousands of responses.

- Quality: a seamless experience with no defects is often worth paying for
- Risk-reduction: mitigating potential problems is difficult to measure, but valuable
- Unique functionality: a capability that no competitor can match is a sensible criterium for a purchase decision
- On-boarding experience: data shows[295] that ease and reciprocity results in higher WTP
- Familiarity: having used a product or a workflow paradigm for years, it is the comfortable way to work

Result: Fair exchange of value. Your product is useful and not excessively painful; the "devil we know;" getting your money's worth; easier to stay than to leave, and no particular desire to leave.

COERCION

- Contract lock-in: retaining your business through paperwork rather than by choice
- Data lock-in: retaining your business by holding your data hostage rather than by choice
- Effective monopoly: being the only feasible option*
- Effective price-fixing: breaking the so-called "free market"

* This can be constructed purposefully, e.g. Uber spending tens of billions of dollars subsidizing rides to drive rival taxi and ride-share services out of business, so that they are the only option, and can raise prices, as they now have done.[296]

- Middle-man: placing yourself in the middle of a transaction, increasing consumer price while decreasing supplier's profit*
- Bundle-stuffing: combining many things the customer doesn't want with the few they do want, to charge more in total**
- Scale Anti-Pricing: raising prices once an installation is at-scale, knowing that although an alternative might be more effective, more desirable, and cheaper, the one-time cost of switching is incredibly high
- Predatory Pricing: using lower-than-cost pricing to destroy competitors and ward off investors (funded by another business unit like Amazon does or by VCs as companies like Uber did[298]), then increase prices once the competitive market has been decimated and customers have no choice.
- Patents: abusing a system meant to temporarily protect inventions to block normal competition.
- Corporate policy: once a product is written into a company's formal policy (site-wide license or the only approved vendor for some application), that product "wins" even if every user hates it
- Government fiat or regulation***

Result: Adversarial. Customers want to leave; they idly comment that they wish some new competitor would arrive and disrupt you; they hate seeing your charges on their bill; they do business with you only begrudgingly; they lobby their boss to switch vendors.

All of these things contribute equally to the economist's definition of WTP: The customer is in fact paying, and might pay more if you raise prices. But strategically they are completely different.

---

* A classic example is the person who buys a bunch of tickets to a concert, then resells them at 10x the price after the concert is "sold out." Here's an even more appalling example.[297]

** There is also a positive version of bundling, in which the items are mostly things the customer *does* want, purchased at a discount over buying each item individually, possibly with some useful interoperability.

*** Here Uber is an example of "love," breaking the "coercive" stranglehold of taxi industry regulations.

"Do me a favor and hit her with some lower expectations first."

credit[269]

EFFECT ON GROWTH AND COMPETITIVE PRESSURE

Love creates inexpensive, non-linear growth, because your customers are your allies.

You get repeat purchases, whether it's a one-time revenue product or a loyal recurring-revenue customer. This creates growth with no additional marketing and sales costs.

You get word-of-mouth advocacy. When someone asks what to buy on Twitter, your rabid fans answer the question. When there's a review site, your product ranks number one. When Customer Surplus is enormous, consumers reciprocate by selling new customers on your behalf.* Once again, this is growth without additional marketing and

* Hollow Knight is a high-quality indie game, made primarily by just three people. Released in 2017, people still make YouTube videos about it in 2023. The sound-

sales costs. Furthermore, the effect grows as your customer base grows: A non-linear effect.

When a new competitor arrives, *even when it is superior* in features or price, your customers will stay, because they're here for more than just the features and the price. This yields retention, which is another form of growth.** There is a limit to this effect of course—at some point the product simply isn't good enough—but it carries you through the vacillations of typical competitive one-upmanship.

Utility helps grow existing customers, and is neutral-to-positive on attracting new ones.

As an organization grows, it will naturally buy more seats of software for teams in customer support, sales, engineering, and so on. It will naturally buy more infrastructure and incur more credit card transaction fees. This isn't a negative, and does creates internal growth, which is a powerful growth-driver for any business, especially at scale.***

But, a customer's willingness to buy another ten seats of JIRA doesn't imply the customer is going on forums, spending personal credibility to advocate on your behalf. And it doesn't mean they won't take a look at an interesting new competitor.****

---

track has millions of listens on Spotify. Everyone says it's far too cheap at $15. Plus you get 4 expansion packs for free—something games normally charge for. Everyone repeats the story about how it's just two guys plus one other guy who did the amazing music. Fans even begged them to charge more but they don't—they'd rather be accessible, and people love them all more for it. The economist would say they should raise prices because they can. Yes they *can*, but it's obvious that rabid fans generate millions of purchases, and that financial impact is so much larger than closing the WTP/Price gap to "demonstrate you have market power."

** Don't believe me? Look at the growth curve (p. 335) of any startup that went from 7%/mo cancellation to 2%/mo.

*** At scale, new customers can be added only so quickly, whereas you have an enormous existing customer base, so growth inside the base is a larger factor than growth from new customers.

**** Indeed, new JIRA competitor Linear has quickly amassed a rabid fanbase on the basis of exceptional UX. It's easy to imagine JIRA users trying Linear and even advocating to switch, whereas it's laughable to imagine a Linear user trying to convince their team to switch to JIRA.

Coercion causes your customers to be *allied with competitors*; they're internally-motivated to leave, so they will.

"Just give me an excuse." Your customers, locked in against their will, cannot *wait* for a viable competitor to appear. They will go out of their way to switch, coming up with reasons why investing in the switch will pay for itself ten-fold, despite the cost. Exactly the case you *don't* want your customers building.

When your contract is up for renewal, you should be very afraid. When someone asks on Twitter what tool they should use, your customers say: "Well we use X, but don't make the same mistake!"

You are constantly vulnerable to disruption, even by mediocre competition. This is the weakest position you could be in, because you're coercing customers instead of delighting them.

"OK, little higher next time. You just gave him a hankering."

credit300

PROFIT DONE RIGHT: CREATE MORE WTP, THEN SPLIT IT WITH THE CUSTOMER

> "[When you increase WTP], you're adding value for the consumer, and then figuring out how to split that with the consumer."
> *–Michael Mauboussin, interviewed on*[301] *the Invest like the Best podcast.*

Creating value for the customer comes first. Then—and only then—you can decide how to "split it with the customer," either leveraging Consumer Surplus for advocacy, high-retention, and growth, or indeed by raising prices.

When you create that value through Love or Utility, this is both sustainable and profitable. When it's through Coercion, it is temporary at best.

The strongest organizations have all three. For example, Apple generates Love through appealing design, being a statement of personal brand, and maintaining the highest standard of privacy even if it means the product is less functional or interoperable. Apple also leverages utility, becoming familiar and convenient (and thus a mental effort to switch away), and trying to become the center of everything from family photos to shared files to 10,000 notes to the common way to purchase things, creating a form of "lock-in" that feels useful rather than evil. But it leverages Coercion as well, as users are locked-in even when they'd prefer not to be, unable to export data from apps like Notes, and being forced to buy new devices as older ones suspiciously stop working well after applying new (mandatory) operating system upgrades, and changing the connectors on charging cables every 5-10 years.

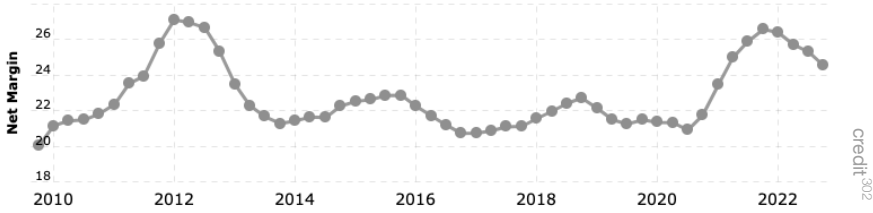


Figure 2: Apple's net profit margin:
If they were increasing prices faster than WTP, profits would have increased.

In any case, Apple has increased WTP in all three ways over the past ten years, and they've split that with their customers, as evidenced by a consistent profit margin (Figure 2).

EVEN THE COLD-BLOODED CAPITALIST SHOULD ESCHEW COERCION

Here's why Love and Utility results in more valuable companies, even though it prioritizes Consumer Surplus over profits:

Imagine there are two companies, alike in every way: Same product, same industry, same market, same number of customers at the same price, at the same costs, and thus the same revenue, same profit, and same WTP. The only difference is:

1. Company's WTP is generated only by Love.
2. Company's WTP is generated only by Coercion.

Which one is most likely to grow in volume and profit over the next five years? Which is more likely to capture more market share? In other words, which is the better investment for a Venture Capitalist?

I'd pick (1). I know their customer base will help them grow efficiently, while competitors look on helplessly, unable to convert customers even with the lure of unique features and lower prices. Whereas I know (2)'s customer base will be trying to leave, praying another competitor comes to save them, publicly warding away potential customers from repeating their mistake.

It is also possible for (1) to add Utility or even Coercive WTP to their strategy, further strengthening their position, whereas it is much more difficult for (2) to generate Love starting from their current position. It's not that Coercion is never an appropriate ingredient, but rather that the other two are better.*

Love beats Coercion, even as cold-blooded, money-grubbing capitalist investor, indifferent to ethics or the betterment of the world.

And yet Love makes money while in fact bettering the world, and making everyone happier.

So choose Love by building it into your strategy, investing in it, and then reveling in what you've created.

---

* It's like the Agile Manifesto:[303] When it says "Working software over comprehensive documentation," it isn't saying "Documentation is bad." Rather, it's saying "Working software" is *more* valuable, so that's where we should spend most of our energy.

APPENDIX: RELATIONSHIP TO OTHER FRAMEWORKS

You can apply this concept directly to your strategy, and merge it with other techniques.

Kano (Figure 3)

"Love" feels a lot like Kano's[304] "Delight"—a joyous, perhaps even unexpected upside. "Utility" maps to "Performance"—where the more of it there is, the more value it is to the customer. "Coercive" maps to "Inverse"—something that customers actively dislike, even though you gain the selfish corporate benefit of retention.

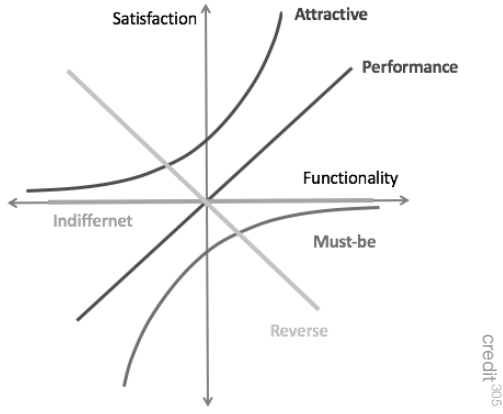


Figure 3: The Kano model

Moats

Many of these things sound like moats (p. 761), and for good reason: Increasing WTP of any type increases your ability to capture and retain customers. The more forceful (whether positive or negative), the more that becomes a permanent advantage that others cannot dislodge. No one can take away a fantastic brand, and government fiat can last for decades.

An interesting example is "network effect," because it shows up in all three types:

- "Love" network effects include community and ecosystem, where participants help one another personally and professionally.
- One "Utility" network effect is a functioning marketplace, so e.g. eBay was for decades the destination having the greater number of buyers and sellers of collectable objects, and thus genuinely the most useful place to transact. You might not "love" eBay, but certainly people went there because it was useful, not because they were forced to.
- One "Coercive" network effect is when choice is limited to "preferred vendors," creating a cartel rather than creating choice. For example, the United States health care system features insurance companies who each support their own network of doctors. A consequence is that switching insurance can mean you have to switch family doctors—an unnecessary and "value-destroying" activity as an economist would say.

Start with "Why"

"Love" reinforces the Simon Sinek's admonition that companies must "Start with Why," (p. 399) i.e. understand and articulate its higher purpose, it's mission, because when that's strong and important, when it permeates everything from its market-positioning to its culture to its employees, it's extremely powerful, and impossible for a competitor to destroy.

Example: Buffer[306] has a relatively undifferentiated product and pays lower salaries than many people can get elsewhere, but their culture and transparency is second-to-none, and people want to be a part of that. Example: TruthSocial, which can't pay salaries like Twitter, and doesn't have the reach of Twitter, and has technical issues with downtime and slow innovation,[307] nevertheless possesses a rabid fanbase because of the mission and community.

Blue Ocean Strategy: The six kinds of "buyer utility"

In Blue Ocean Strategy,[308] W. Chan Kim and Renée Mauborgne highlight six ways in which you can deliver "value" to customers. These are a subset of the more general reasons why people are compelled to buy, but it's useful to emphasize the cases where the customer is benefiting directly:

Blue Ocean Buyer Utility	WTP Category	Commentary
Customer Productivity	Utility	This category is too broad; it is important to distinguish between "more value" and "less cost." Both contribute to "productivity," but it is an order of magnitude more important to increase value (p. 165). It's also important to define value (p. 259).
Simplicity	Utility	Included above.
Convenience	Utility	Included above, in several forms; for example "location" is a specific kind of convenience.
Risk	Utility	Included above.
Fun & Image	Love	Included above.

Blue Ocean Buyer Utility	WTP Category	Commentary
Environmental Friendliness	Love	Included in a more expansive "social and environmental impact," as nowadays (2023) it is more common for customers to make buying decisions on factors like Fairtrade,[309] or purchasing from local or minority-owned business, or supporting businesses with specific values and public commitments, in addition to the idea of being friendly to the environment.

Long-term engagement metrics

Many products wish to "drive engagement." Some point to Facebook as the pinnacle of "growth hacking," driving up numbers, often slipping away from Utility (to say nothing of Love) and into Coercive tricks.

But even at Facebook, solving for Utility over Coercion worked better. In a fascinating multi-year UX experiment,[310] Facebook found that when they reduced the quantity of notifications (by keeping the quality high), it had the expected negative result on engagement: Customer satisfaction increased, but app usage decreased (because it was leading you back to the app less often). But, after a year, app usage actually increased and remained higher that it was before the change. Increasing genuine satisfaction created more engagement in the long run. They had to be patient to see the results; traditional "growth hacking" did not discover the best solution.

Many thanks to John Doherty[311] *for contributing insights to early drafts.*

Chapter 15:
How startups beat incumbents

ADVANTAGES AS WEAKNESSES
UNQUANTIFIABLE RISKS · NICHE · UNSCALABLE
SERVICE · NEW TECH · DRASTIC CHANGES
OPINIONATED · POSITIVE-SUM · WORSE
NEW PROFITS

It doesn't seem possible for a startup to beat an incumbent.

An incumbent has everything: money, brand, customers, a sales team, marketing that generates thousands of leads every month, product and engineering teams that constantly ship. They mine their big existing customer base for ideas, and then build exactly the right features, and then charge for it. Their 24/7 support team provides faster and better service than someone working in their pajamas at home. They don't have to build the basics or ask Twitter how to manage international sales tax. They can just focus on innovating.

Of course if you've ever worked at a big company, you know that **while most of those things are true, it doesn't feel like it.** Big companies are rarely well-oiled innovation machines, and it certainly doesn't feel like you're constantly outpacing the competition.

When we analyze how incumbents are vulnerable, we uncover opportunities that startups can exploit to win, where there's often nothing the incumbent can do about it, despite their advantages:

- Taking risks that cannot be quantified
- Addressing a profitable niche
- Doing delightful, valuable things that don't scale
- Unsurpassed customer service
- Leveraging new technology
- Make drastic changes
- Having an opinionated personality
- Doing things that aren't zero-sum
- Being worse-but-acceptable in most dimensions
- Being low-cost against a profit center

THE PATTERN: EVERY BIG-COMPANY ADVANTAGE CREATES EXPLOITABLE WEAKNESS

The reason big companies don't function as well as described above is that things at scale are super-linearly more difficult (p. 773).

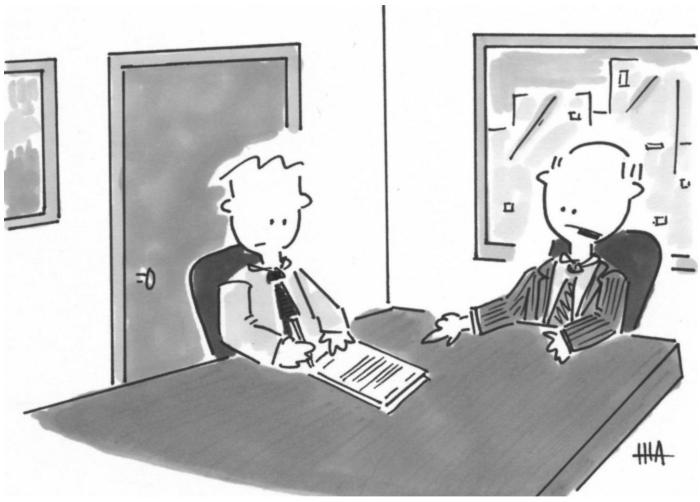
It's an advantage to have 100,000 customers when you're figuring out what the next feature should be, or when you're launching a second product, or when you get free growth from word-of-mouth.

But it's a disadvantage to have a lot of customers when you want to innovate with your product, because no customer wakes up in the morning and says: *Gee, I hope the software I'm accustomed to dramatically changes today*. Customers don't want to learn new UIs. Customers have workflows that you have to accommodate. Old technology that powers those 100,000 customers doesn't support the latest technology. You have to update documentation and videos and the people in support and sales who need to be retrained. Even a simple change can be difficult and expensive,[313] and certainly low-ROI.

Besides "scale," a big company must accommodate things startups can ignore.

There's the legal department, for example. A startup does all kinds of illegal things. Most startups do not pay taxes properly, sometimes not at all, especially in other countries. Startups don't adhere to all the Acceptable Use Policies of all the products they use. Startups don't have a security team who vets vendors before sending them sensitive data, or vets libraries before they're integrated into the code base, causing all of their supposed "secret intellectual property" to become open-source.

As a result, the startup not only moves more quickly—which is how most people characterize it—but they can completely skip things that a larger company cannot. So Uber decided to just do illegal things in order to grow. An incumbent taxi company obeys the law, so they lose.



credit[314]

"This just says you won't reveal anything about our nondisclosure agreement."

You could say that that's not fair. You could say that's what regulation ought to prevent. But the reality is that startups often ignore the law, and that can be an edge.

The way a startup wins, is to do things that incumbents cannot or will not do.

So, let's see how to attack where they cannot defend.

TAKE RISKS THAT CANNOT BE QUANTIFIED

The way a larger company decides to take a risk, such as launching a new product line or entering a new market, is by creating a detailed analysis of the opportunity, and a cost estimate. Then the decision is:

1. Is this is a good ROI? *(potential-revenue divided by costs)*
2. Do we have conviction that the risk of failure is low?

How can a startup exploit this decision process?

Starting with decision (1), the analysis is typically wrong. There are studies everywhere—and your own experience, if you've worked at a large company—showing that most development projects are significantly late and over-budget, and also that the outcome is typically worse than expected. Both sides of the ROI fraction are worse.*

So, whichever projects *appear through traditional cost-benefit analysis* to be low-ROI, are unlikely for an incumbent to do, even though there's a good chance that (a) they're rejecting genuinely good ideas and (b) they're accepting weaker, more straightforward ideas, only because those more readily lend themselves to ROI analysis. A startup who selects "apparently" low-ROI projects, will probably have no competition from incumbents.

With decision (2), big companies don't like to take big risks *even when the outcome might be large*. The fear of failure is an order of magnitude more motivating than the desire to innovate or even the greed of success. One reason is that the core business is probably going well, and you "don't want to mess that up." Another reason is that no one wants to be the one who proposed, fought for, and then presided over a multi-million-dollar failure. Another reason is that none of us really knows what a "big risk" is anyway, because even experts can't predict what will happen (p. 193), and we don't know how to talk about risks (p. 997) or measure risks (p. 1319), even in retrospect (p. 1253).

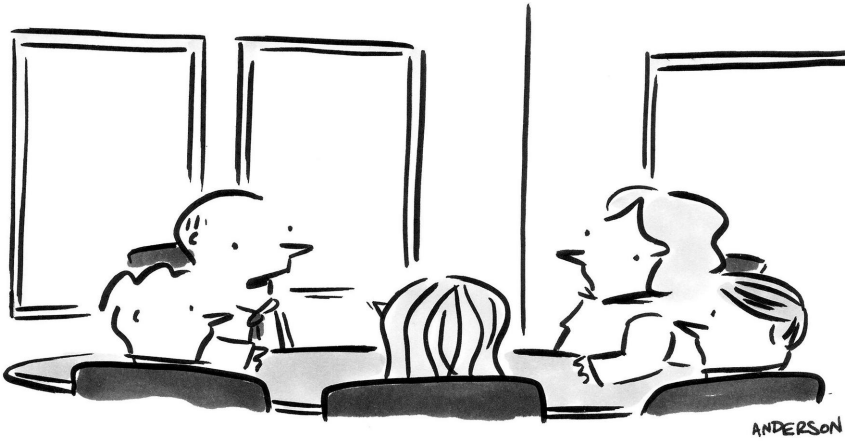
At a big company, it's safer to say "let's gather more data" and "let's wait for consensus" than it is to take a risk.

But innovative things are often high risk or unknown risk.** Therefore, a startup can pick things which are risky, or where the risk

---

* I cover exactly what to do about this in an article about how to do ROI analyses correctly (p. 171).

** "High risk" means we know number that is the probability of success, and it is low. "Uncertainty" is unknown risk, meaning we don't know what the number is



credit[315]

"OK, now that we all agree, let's all go back to our desks and discuss why this won't work."

is impossible to ascertain, but where the potential upside is high, and incumbents are unlikely to follow.

Indeed, this is also what I recommend for work-planning in general in selecting and prioritizing Rocks versus Pebbles (p. 221), where the "big Rocks" that move the needle (p. 1065) should be selected on the basis of potential outcome rather than ROI, whereas the Pebbles should indeed be based on ROI.

Startups can do it, while incumbents are almost always too fearful.

That's OK, the incumbent can buy you later at 10x revenue.

---

at all. Though different, they have the same result in this context: It's too scary to take the chance.

ADDRESS A NICHE

An incumbent wanting to expand into a new market or launch a new product line, must apply a significant amount of money and people—a large investment, even for them. What kind of return do they need on that investment?

The answer is: It has to materially affect their growth rate.

The rule of thumb is it should increase their overall revenue by at least 10%; some people call this "the materiality threshold." However, that number goes up as the (perceived) risk or (actual) investment increases. For an incumbent with hundreds of millions in revenue, that means the product line must have a good chance of making $50-$100 million, or they won't even try (nor should they).

> *"Given our size, we only see a few good things [to invest in]. If we were smaller, then we'd see lots of good things."*
>
> *—Warren Buffett.*

Another financial metric that creates a materiality threshold is a certain ratio on the Profit & Loss statement.

A software company at scale spends[316] around 20% of their revenue on R&D, which includes Product, Engineering and Design. A fully loaded team, including salaries, taxes, software, hardware, training, management, travel, office space, and so on, can cost upwards of $2 million a year.* If that cost is supposed to be 20% of revenue, the

* Obviously this varies by geography, employment laws, tax laws, and the size and composition of the team; this estimate assumes a team of eight engineers, a prod-

team needs to generate at least $10 million in annual revenue, which means the product they're working on must do that, even if the product is a small add-on or the team owns a subset of a larger product.

Of course, a small startup doesn't see it that way. A team of two founders and one employee isn't thinking "We have to make $10 million a year, otherwise it's a failure!" This means the startup can focus on a niche that doesn't need to generate $10 million; it could generate $1 million.

The startup can focus on a niche and ICP (p. 317) that a big company cannot afford to target, either because that niche wouldn't hit the overall revenue materiality threshold, or wouldn't hit the one-team P&L threshold.

With that focus, the startup has no direct competition from an incumbent. In fact, the larger the incumbent, the less the startup has to worry about competition.

Feels good.

DO THINGS THAT DON'T SCALE

WP Engine was the first in our market to support LetsEncrypt.[317]** We knew nearly all of our customers would want it. We wanted to promote it heavily, but we were already deep into scale, with 100,000 customers who could potentially use it on day one.

---

uct manager, a designer, and a manager, in the US, and rounding-off for rhetorical simplicity.

** If you're unfamiliar with this web technology, don't worry—the details aren't important to the story. Suffice to say: It was a desirable capability, and is now ubiquitous.

Therefore, it had to be scalable before we released it. That means breaking down the components into queues, in case one step in the process was faster than another, or in case one failed and had to be repaired before the system could make progress. And we had to monitor those queues, and send alerts to humans if it stayed broken for too long. And we had to run at-scale tests to make sure it worked when there were 1000 simultaneous requests with random failures. And we had to train hundreds of folks in tech support on the questions we anticipated, and train hundreds of folks in sales on how to leverage this to make sales, and work with marketing on how to message it. And we had to make sure it had close to zero bugs, because if thousands of people start using it, and 10% ran into a bug, we'd crush our support team, and hundreds of people would take to Twitter to complain.

We were correct to invest many months of time in all these areas; on the first day, thousands of people *did* start using it, tens of thousands in the first month, and indeed some components did break, and lots of people asked questions in tech support. And people praised us on Twitter as a result.

The good news about a large customer base, is that you can have 1000 users on day one and 10,000 on day 30. In this case we gave it away for free, but in general an incumbent can be generating $1M ARR or even $10M in short order. A startup cannot do that.

But a startup can launch something in a few weeks and iterate. We had to be heads-down for most of a year. And so another startup opportunity emerges: A startup doesn't have to operate at scale.

You might say: A larger company can still make an alpha version of a product, show it to a few dozen customers, and iterate from there. Indeed, we also do that; it's a wise process. However, what a larger

company is *not* willing to do is take the next couple of years just to get to $1M ARR and then take a couple more years to get to $5M and then take a couple more years to get to $20M. That's just way too long to get the "material" amount of revenue and does not leverage their at-scale assets.

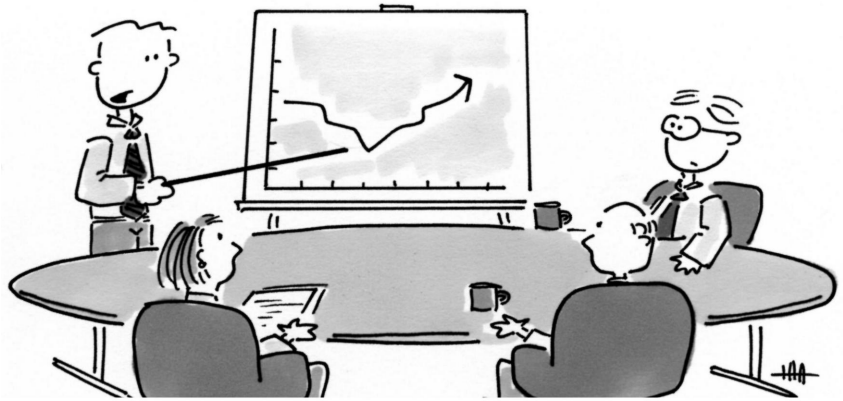
But the ramp I just mentioned is ideal for a startup; in fact, that is a highly successful growth rate. It allows time for the product to settle in and slowly get to the point where it is scaling. This is a reasonable, fun, and plausible path. No one knows you exist; that's bad for sales, but good for iterating without harming your reputation.

Therefore, if a certain product idea is naturally very easy to scale from day one, that's easy for an incumbent to copy. But if the product is naturally difficult to scale from day one, that's ideal for a startup.

UNSURPASSED CUSTOMER SERVICE FROM FOUNDERS & ENGINEERS

I personally handled support tickets every day at WP Engine until we had around 35 people; this is typical for a customer-oriented founder. There's no better way to understand how people interact with the product than to talk to them about when it's going wrong or not meeting their expectations. You get feature ideas, you understand how to streamline the product and how to increase retention.

Customers are impressed by the quality of support and the range of problems you can solve. They won't get that from a company with a thousand tech support reps. Some startups aren't interested in providing great support, but those that do are naturally and even effortlessly orders of magnitude better than a large incumbent. It's a competitive advantage available to everyone.



"So, as you can see, customer satisfaction is up considerably since phasing out the complaint forms."

credit[318]

As if those benefits to both customers and product development aren't enough, it also fosters real love and loyalty from customers. That love translates to forgiveness when you do have problems; see the mountain of supportive tweets when a small-but-lovable company has a big outage or security issue. It also translates into word-of-mouth advocacy, as customers naturally reciprocate, and thus great support results in inexpensive growth. Love is the best form of "willingness to pay." (p. 275)

Which incumbents cannot compete with.

As a startup scales, it loses this advantage. I distinctly remember each time in the past 15 years at WP Engine that a new competitor would brag about their amazing tech support. While WP Engine continues to objectively* have world-class support, it's not the same as personal attention from the founder of the company!

You could decide to never grow that large, optimizing for profit and efficiency rather than revenue and scale, and make Support a permanent competitive advantage.

---

* Won a dozen Stevie Awards, and maintains 98% CSAT.

Max Lynch ✓
@maxlynch

I think we won a lot of deals @Ionicframework largely because of this. We were tiny relative to a lot of the vendors our customers used but we had considerably better customer experience. This is your advantage as a small company, gotta use it

credit[319]

No matter which future you pick, this is still a great way to get started.

LEVERAGE NEW TECHNOLOGY

When a technology is new, the risk of using it is high.

Maybe it won't be supported in five years. Maybe you won't be able to hire dozens of engineers who are familiar with it. Maybe* it will have a big security problem. Maybe it works well for the "Hello, World!" case, but doesn't work at scale. Maybe it's efficient for one developer but too difficult to coordinate with thirty. These are all reasons why incumbents are absolutely correct in avoiding new technology.

But a startup doesn't have these concerns. Not because the big company is wrong, but because the constraints are different. The thing that will kill the startup is not going to be the tech stack; it's going to be that it's too hard to find customers, or they don't have a budget for this problem, or it's too hard to compete, or you run out of money, or any of the other things needed to get to Product/Market Fit (p. 9).

* In fact, *certainly* it will have *many* security issues... the question is whether they will be identified and addressed.

And new technology often creates a competitive advantage. New technology makes certain things efficient, or enables things which previously were impossible, as in the current case of AI (p. 419).

The startup *is* taking a risk on that technology. If you're banking on a new open source project, and it doesn't take off, you might have a product built on a platform that is no longer supported, and that's bad even if you're a small startup.

However, what is the worst case, assuming the startup isn't already dead by then? It's that, five years from now, you've built a sustainable company, and now you have to redo your platform using different technology. That does really suck for you. You might have to pause new features for a year to make the transition, and engineers, product managers, and sales reps alike will hate that. But if this penalty "buys" you a successful company, then it was worth it.

MAKE DRASTIC CHANGES

Of course it's nice to have 100,000 paying customers, but it also means all change is difficult, expensive, and risky.

You can't change the user interface even slightly without a torrent of angry tweets, confused people clogging tech support, and updating all documentation and screenshots. You have a solidified brand with expectations on positioning and pricing; changing any of those is not only disruptive, but risky. On the back end, you've optimized and built for scale, which means you're locked into architectural choices, technological integrations, and business partners.

New startups can change any of those things, even drastically, even if they have 100 customers. Obviously you'd prefer to have more customers, but so long as you don't, take advantage of the fact that you pivot into the market reality and your own strengths. Especially when

the market shifts—like today with AI (p. 419), and yesterday with mobile devices—the incumbents are the ones who cannot quickly adapt to the new world, whereas the little startup can adapt immediately.

Part of why "big companies are dumb" is because they can't change even when they know they're doing something dumb. New companies can.

HAVE AN OPINIONATED PERSONALITY

It is rare for a large company to express a personality.

There are many reasons for this. They want to address a large and therefore diverse market to sustain their revenue and growth; by speaking to everyone, they speak to no one in particular. They have an established brand, which creates trust, which is one of the reasons they win sales, embodied in the now-outdated phrase "No one ever got fired for hiring IBM." So the language on the homepage and inside the product needs to reinforce this trust, which means being impersonal but solid.

Another reason is that customer communication is spread over hundreds or thousands of people, from Support to Sales to Marketing to Product to Design. There isn't such a thing as a genuine, human personality and style that a thousand people share and express identically. Whereas everyone can conform to generic but professional language.

But a startup doesn't have these constraints. Indeed, the founder often has a strong personality, with specific ideas of what's good and bad and how things should be done and how to express it. That's at least partially why they started a company in the first place.

Some potential customers will be attracted to that personality and some will be repulsed. But that's true of anything that is wonderful and different and powerful in this world.

You're a little company, now act like one.[320] In fact, you will directly win customers (p. 939) because of it.

When a large company tries inject personality, it often comes off as contrived, not genuine. Whereas with a startup, it feels genuine because it *is* genuine.

Some customers only want to buy from the market leader. That's rational, and if that's a primary deciding factor, there's little a startup can do about it, no matter what their home page says. Therefore, the startup should ignore that segment and focus on customers who want to buy (p. 317) from a plucky upstart that has something to say.

DO THINGS THAT AREN'T ZERO-SUM

Some things in business are a zero-sum game.

In a zero-sum game, when one player wins, another player necessarily loses. Poker is an example: When one player wins chips, other players lose exactly that number of chips. Blackjack is a counter-

example: Players at a table individually win or lose, unaffected by other players; no player loses chips when another player wins.

Marketing gives us examples of each. Zero-sum marketing channels arise when there's a power law[321] or where there is exclusivity. Examples:

- SEO *(The top positions generate more traffic than all other positions combined)*
- AdWords *(The top positions generate more traffic than all other positions combined)*
- Affiliates *(The top few affiliates generate more leads than all others combined)*
- Retail shelf space *(Limited surface area)*
- Exclusive distribution deals *(The zero-sum game is created by agreement)*
- Government fiat *(A vendor can be written into law)*

Conversely, there are channels that are non-zero-sum, and more to the point, where even a well-funded, strongly-entrenched incumbent cannot prevent others from winning:

- Social media *(anyone can create a great social presence)*
- Newsletters *(anyone can create great content marketing)*
- Collaborative promotions *(both players more money than they would have)*
- Ecosystems like Salesforce and the Apple Store *(all players make more money)*
- Consumers who buy multiple products *(e.g. 3D animators often use multiple tools)*

Market dynamics can also create either type of game. Stagnant or shrinking markets are zero-sum; without new customers entering the arena, winning one customer means a competitor cannot have that revenue. In growing markets there's a steady stream of new dollars

from new customers, so many competitors can grow. (This is yet another reason why startups should target growing markets, whether the goal is to build a small, profitable company or a unicorn.)

Incumbents are stronger in zero-sum games, because they can apply money and specialized expertise. They can even over-spend, because their scaled business model can absorb a low ROI activity, intentionally losing money in order to stop competitors from using that channel. This doesn't mean startups should *never* play zero-sum games, but they are more difficult, and sometimes impossible.

But incumbents cannot stop startups from winning non-zero-sum games, so that's where a startup should invest (p. 867).

BE WORSE BUT ACCEPTABLE IN MOST DIMENSIONS

No one wants their website to go down.

It's surprising how hard it is to keep a website up for an entire year. For example, "99.9% uptime" might sound excellent, but that equates to 44 minutes of downtime every month! If our company WP Engine had even close to that much downtime, customers would revolt, and rightfully so. Yet, most hosting companies* promise only 99.9%–99.95% uptime.

The reason is that it's very hard to push the number higher. It takes an order of magnitude more direct spend, software development, and processes to get to 99.99%, to say nothing of the proverbial "5-9's"[322] that industrial operations sometimes target. Every little rare, strange, unpredictable thing will knock you out of compliance; it's expensive

* Including the major cloud providers, managed WordPress platforms like WP Engine, and specialized providers like Toasttab for restaurants

and difficult to solve all those cases. And yet, 99.99% just doesn't look that different from 99.95% on the pricing page.

Incumbents, however, often do have to invest in optimizing these expensive dimensions like. One reason is that at scale, rare things become common (p. 1345); at scale you have no choice. Another reason is that it can win sales in some segments of the market; at WP Engine we have enterprise customers who have internal policies demanding 99.99% uptime, so we win against competitors with lesser guarantees.

Startups don't have the rare-at-scale problem, and they can choose their target market (p. 317) such that customers don't have extreme demands,* and therefore startups can win while incumbents labor.

More examples where incumbents have to care, but startups don't:

credit323

"Kathy, if you agree to these terms of service, click 'I do.'"

* Indeed, startups should actively avoid those customers. A startup with a new product definitionally won't satisfy customers with myriad, difficult demands.

Security
WP Engine spends tens of millions of dollars a year on security, ranging from internal teams to corporate policies to annual employee training to SOC and ISO compliance to software reviews to vendor security reviews. For us, security is one of our main selling points, so this is important both for scale, brand, derisking, and because it's what customers pay us for. But that's not true of most products, and most customers don't demand it.

Quality
If the entire product is low quality in every dimension, that's just a bad product. I doubt anyone is excited to build that, not even as an SLC (MVP) (p. 101). For example, uptime is important for WP Engine, but for a SaaS product that is used only during normal working hours, targeted at a certain geography where "working hours" is a well-defined time frame, having even an hour of downtime in the middle of the night doesn't affect customers at all. A large, global company doesn't have that luxury.

Scalability
If a product will never need to handle "big data," then the product can be built with all sorts of simplifying assumptions* that make development faster, safer, even more enjoyable. The UX can be simpler if the users have basic needs, as opposed to nested security groups driven by an external LDAP service.

Compliance
Large companies accumulate internal policies. These are for good reasons, like ensuring that not everyone has access to all data (especially customer data), ensuring that IT teams are capable of managing and upgrading thousands of devices, safeguarding ownership of their intellectual property, and enabling sales to large enterprises and governments in various verticals who impose policies on their vendors. This is one of the reasons they can get multi-million-dollar three-year contracts

* PostgreSQL instead of BigQuery, SQLite instead of PostgreSQL, Python instead of Erlang, reading into memory instead of streaming, batch jobs instead of assemblages of queues and auto-scale groups, standard algorithms rather than distributed computation, normal debuggers instead of distributed logs, off-the-shelf libraries and vendors instead of bespoke solutions

and the start can't. But it also severely hampers what they can do, or at what speed they can do it, or at what cost, and therefore at what customer-facing price.

Legal
We covered this earlier. While I would never advocate for startups to do illegal things on purpose, it's a simple fact that startups often (unknowingly) don't comply with all laws. It doesn't affect their sales; in fact, it might increase it.

Most customers don't care about most things. This is great news for startups, who can select one or two dimensions to care about, and the ideal customer segment who also cares mostly about those specific things (p. 317), and win that segment while incumbents chase complexity in all quarters.

While incumbents have to charge more to cover the costs of multi-dimensional excellence at scale, a startup can charge less for a product that's objectively "worse" along many dimensions, and thus the startup can win on price and still be profitable.

Startups can be worse, but unique (p. 891), and better where it counts.

BE LOW-COST AGAINST THE PROFIT-CENTER

An incumbent cannot change its business model.

The assets that give the incumbent its advantage are also static constraints. The brand is entrenched in consumers' minds. The software relies on platforms and languages and libraries that cannot be changed without a massive rewrite which infamously almost always fails. The business model of marketing, sales, service, and profits is set; a company with the costs of a global sales team, the white-glove on-boarding

team, the expensive infrastructure, the vendor costs, and shareholder expectations that profits will only increase, cannot drop prices.

Therefore, an incumbent cannot compete with a startup which is "a 'lite' version for 1/10th the price."

This is a softer way of restating Disruption Theory,* in which incumbents see the startup coming but, rather than compete directly, reposition themselves to focus on their best, most profitable customer segments, thereby allowing the startup to thrive.

More specifically: Whatever generates the most profits for the incumbent, is the thing they are least able to change.

You have to be careful, because incumbents will spend a lot of money and attention defending their profit centers from attack. But price won't be how they defend. That defense means you should leverage other topics from this article; for example an incumbent can decide to spend "too much" money on AdWords to make that a worse channel for you, but they cannot stop you from having a great content marketing strategy.

Incumbents are strong in most ways, but they are vulnerable.

The only mistake is for a startup to go head-to-head with an incumbent where the incumbent is strong.

Attack where you are strong (p. 543), and they are weak.

This is how to build a winning strategy (p. 489).

---

* Famously explained in Innovator's Dilemma,[324] the theory is more specific than what I'm saying here, involving new technology that is "worse, but cheaper, but in some ways better," where the incumbent seems to act rationally but ends up being completely disrupted.

Harry Stebbings
@HarryStebbings

I hate competitive markets. Being one of 20 is not interesting for me.

I want to compete against:

1. Stale, old incumbents,
2. built on outdated tech infrastructure,
3. with terrible talent brands,
4. lazy product roadmaps
5. arrogant approaches to customer service.

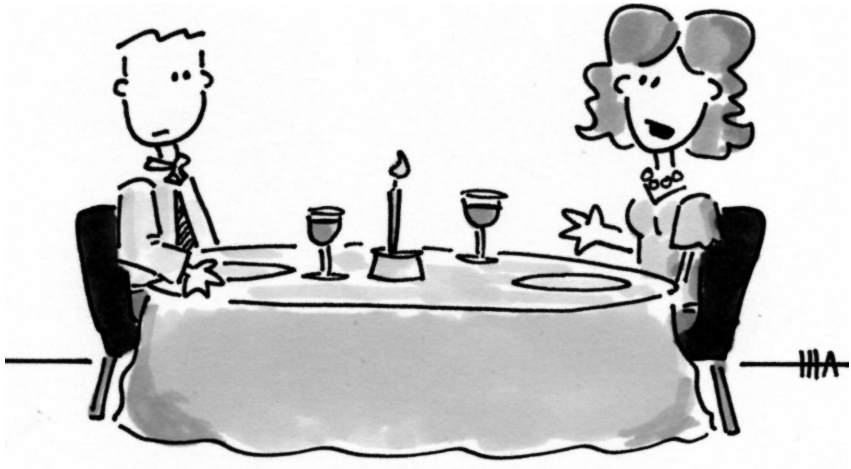
1:13 PM · Mar 9, 2024 · **34.4K** Views

22 25 199 139

credit[325]

Chapter 16:
Selling to Carol: Why targeting an ICP brings 10x more customers than you expected

EVERYONE IS NO ONE · SELLING TO CAROL
SURPRISE · PRACTICE · FINDING CAROL · REVENUE



"It's not that I don't like you, Ted, you just don't fit my target demographic."

credit326

SELLING TO EVERYONE MEANS SELLING TO NO ONE

Everyone goes to your home page, therefore your home page has to speak to everyone, right? Buyers, users, existing customers, curious explorers, all working at companies large and small.

Most companies approach this the wrong way, which is to speak to "everyone." The worry is, if you were to speak only to a narrow customer segment, everyone else feels excluded or confused, and will bounce off the home page and buy something else. So you end up with generic positioning statements like "The Power To Know." (Figure 1)

SAS has trademarked "The Power To Know®" but I'm not sure why. Can you tell me what the product does? Who it's for? Did you read that entire paragraph? Surely not.

Figure 1: SAS is The Power To Know®, but know what exactly?

Could anyone in the world—even their perfect, ideal, best customer (ICP: Ideal Customer Profile)—be intrigued and excited to learn more? If that ICP happened across the website, would they even *know* they are the ideal customer?

The first step in disabusing yourself of this fluffy language is to see that by *not* speaking to a specific customer, you're saying *nothing* to *everyone* and thus *everyone* will bounce off the home page, and never think of you again. (You didn't give them anything to think *about*.) If your ideal customer doesn't know you're selling to them, who will?

By the way, The Penny Group also delivers "The Power To Know®," also trademarked (Figure 2). So much for defensible differentiation through trademarks.

Generalized messaging has no power, no emotional connection, no interest, no information.

Exactly the opposite of what you need from your advertising or home page.

The next step in disabusing yourself of this idea of "speaking to everyone" is to acknowledge that it is impossible to speak to everybody at once. You want software developers to know you have an extensible API, but you want the corporate buyers to know that your product makes them more money (p. 165). You want to look professional so that managers at large companies are comfortable trusting your company, but you don't want to so aloof that small company buyers see you as "too corporate" and can't relate. You want to highlight configurable workflows and reports that allow large customers to apply your tool

Figure 2: So much power! So much knowing!

to all their teams, but you want small customers to realize that you can turn all that off so things are simple and easy.

Once you agree with these two points—you must be evocative and specific, and you can't speak to everyone—you must conclude that you have to speak to just one type of customer.

But how do you pick, and how you avoid excluding everyone else, reducing your target market to a tiny niche?

We'll start with the first question.

SELLING TO CAROL

Describe a perfect customer. We'll call her Carol. Pick a concrete company that she works for, a company similar to one of your existing, thrilled customers. What's her official title and what does she do? If your potential market includes a wide variety of company types and positions, *just pick one in particular*. Whatever problems your product solves, Carol has *all* those problems. She has all the problems, she knows she has the problems, she already has the budget to spend on those problems, and she's already looking for how to spend it, just like it says in your market analysis (p. 71). Write those problems down from her point of view, the way she would describe them if complaining to a friend over lunch. Whatever advantages you have over your competitors,[327]* Carol needs exactly those things. Whatever makes people love you,** Carol has those attributes. Whatever makes people

---

* Not sure? Ask your Sales team—they'll tell you, "oh yeah, any time someone says _____, I know we're going to close the sale."

** Not sure? Ask your Support team—they'll tell you, "oh yeah, the best people are _____ because they ask great questions and they're always telling me how much they love us, even when they have a problem."

continue paying for the product forever,* Carol has that. List all these things.**

Carol is literally custom-built to be blown away by your product.

Now the question is: What would a web page / Google ad / print ad / trade-show booth / postcard be like, such that Carol would immediately understand that you are her savior? Remember, you get only 3 seconds to grab her attention and another 5-10 to convince her that your product is the second coming.

Can you make it clear in a picture? Maybe a before/after she can relate to? Will describing three features make it plain? Will pointing out your best competitive advantage make her weep for joy in finally finding a company who "gets it?" Can you ask a provocative question, something she identifies with? Is there a phrase where she'd laugh out loud because "that's so true?"

You only get a few seconds, so a paragraph won't do. You have to communicate in a picture and a few words. The good news is you have to please *only Carol*, and you know Carol. You even know she'll honestly be thrilled to find you.

If your ad can't grab Carol's attention—your perfect customer—why do you think it will grab anyone else's attention?

If you still say it's impossible to communicate your message in 5-10 seconds, *no one in the world will get your message*. Simplicity is as important in positioning as it is in strategy generally (p. 489).

---

* Not sure? Look at your retention data—which customers stay forever, even if you think they shouldn't? Not sure about that either? Use the Iterative-Hypothesis method to interview them (p. 239)—you need to do that anyway so you can build great products.

** Some pundits recommend inventing a personal back-story like "Carol has two cats and loves Mahler." Don't do that. It's irrelevant and inaccurate. Stick with professional attributes, what Carol is interested in, why she is interested in it, why she is looking for a new solution right now, how she makes those decisions, what her budget is, and so on, all topics listed in the aforementioned Iterative-Hypothesis method (p. 239).

Figure 3: Twilio targets developers, even though the buyer isn't a developer. This campaign was so effective, it turned into an entire book.[328]

When you're willing to speak only to your target audience, it is incredibly powerful (Figure 3) (Figure 4) (Figure 5) (Figure 6) (Figure 7).

SURPRISE: YOU'LL AUTOMATICALLY SELL FAR BEYOND CAROL

Now let's tackle the fear: That targeting Carol means you're excluding the rest of the market, and therefore limiting your growth and reach, even if you perform well in your niche.

That's not what happens. Here's what actually happens:

You have a target market—the bullseye—and the center of that bullseye is Carol (Figure 8).

Every product has strengths and weaknesses. For Subaru:

Figure 4: Dollar Shave Club was unabashedly "cheap but good," and didn't care about sleek advertising like Gillette. It got everyone's attention because it was such a clear message, and sold razors to everyone, not cost-conscious consumers only.

Strengths	**Weaknesses**
• Low-cost	• Low performance
• Reliable	• Not sporty
• Safe *(top-rated by government)*	• Not cool
• Rugged *(take it off-road)*	• Ugly

Carol is constructed such that she values everything on the left, and actively does not want anything on the right. To her, those aren't weaknesses, those are actually strengths, because she wouldn't be caught dead in a Porsche, her lifestyle isn't sporty, and she disagrees with your definition of "cool" and "ugly." (Figure 9)

But Carol isn't the only person attracted to this set of trade-offs. There are other people who either like the things on the left or are at least indifferent to them, and are indifferent to the things on the right. For example: I drive my daughter to school, drive to work, and spend a lot of time on highways and residential roads, so I want some-

Figure 5: Subaru knows it isn't cool; it highlights its positive attributes.

Figure 6: Subaru is about accessible, safe family life.

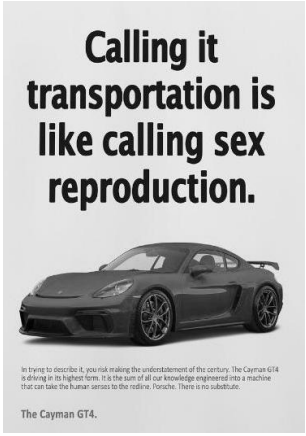


Figure 7: Porsche's message is… differentiated from Subaru's.

Figure 8

thing safe and reliable, and I don't care about high performance or being sporty.

That's the next ring in the bullseye—people who generally value your positives and aren't dissuaded by the weaknesses. **There are 10x more of those people than there are of Carol.** Let's call that person Diana (Figure 10).

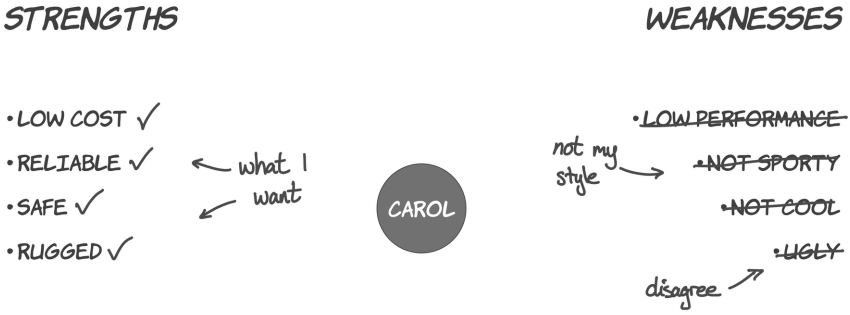


Figure 9: Carol's preferences

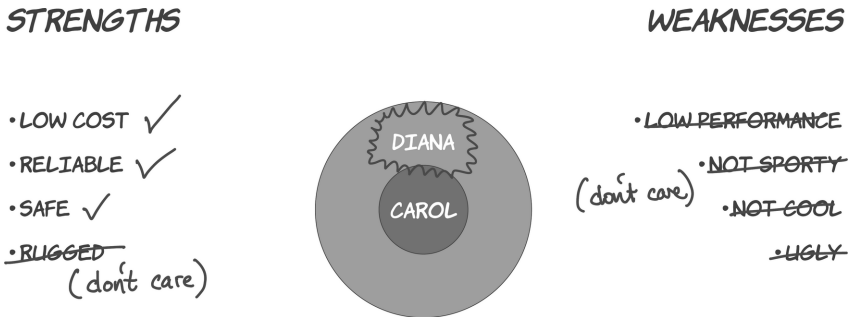


Figure 10

But that's not all. Buying a car—or anything—is always a set of trade-offs. You rarely find a product that fits your preferences perfectly, i.e. you're rarely The Carol for any product (or even The Diana). So, you weigh the pros and cons. Maybe I don't care about being rugged because I don't go off-road, but I do care about safety and reliability, and I'm neutral on cost. And on the weaknesses, I'm disappointed that it's low-performance, and I do care what it looks like. But, I don't care about "sporty" or "cool" specifically.

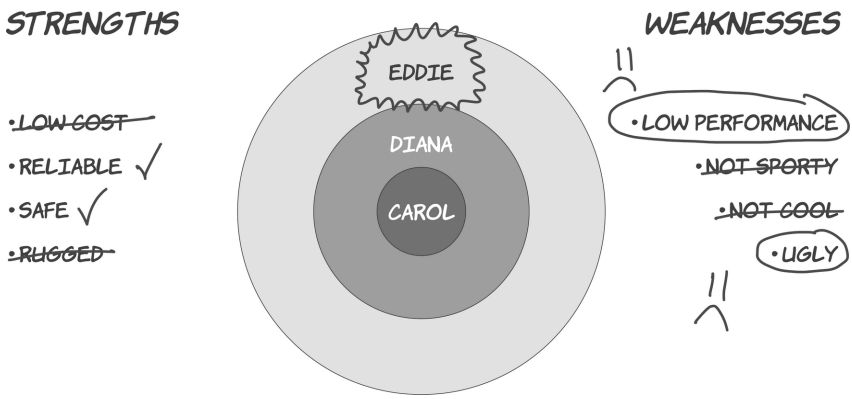


Figure 11

So where does that leave me? Who knows—everyone will weigh these things differently. Clearly *some* of those people will still consider the Subaru to be the best-possible set of trade-offs, and **there are 10x more of those kinds of people than there are of Diana.** Call them Eddie (Figure 11).

This is why, contrary to your fear that targeting Carol dramatically limits your target market, your effective target is actually at least 10x larger (because of Diana) and more like 20x-100x larger (because of Eddie).

Targeting Carol makes your message clear, compelling, evocative, even emotional. But trade-offs are how people buy, and exactly because you were so clear about what those are, you've paved the road for many people to make that choice.

DOES THIS THEORY WORK IN PRACTICE?

Continuing the Subaru example,* the advertisements above are about being low-cost but practical. These are attributes valued by many people. But in the 1990s, sales were in decline, so they researched which consumer segments were most likely to buy a Subaru. They found five categories: Teachers, healthcare professionals, IT professionals, "outdoorsy types," and lesbians. Lesbians were the strongest category—four times more likely to buy.

Targeting lesbians was a risky move in the '90s. There were still no mainstream TV shows or movies with gay characters, and celebrities were still in the closet. When Ellen DeGeneres (both a celebrity and star of a popular TV show) came out as gay in 1997, advertisers pulled out, including Chrysler. When Ikea ran a commercial featuring a gay couple, their stores received bomb threats.

But as Warren Buffett says, "Be greedy when others are fearful," and Subaru invested where others feared to tread (Figure 12) (Figure 13).

The campaigns were a hit, but not only with lesbians. Subaru grew faster in the subsequent ten years than any other car company,** and while the gay and lesbian demographic remained their strongest niche, the vast majority of people who bought a Subaru were straight. And they were buying the car because they value its attributes—affordable, safe, rugged—and because on balance they cared more about those than the attributes it lacks—sleek, sporty, high performance.

---

* This story is expanded and cross-referenced in Alex Mayyasi's 2016 article in Pricenomics.[329]

** Measured as a percentage of total sales volume[330] from 1995 to 2005, of car companies that were founded before 1990.

Figure 12

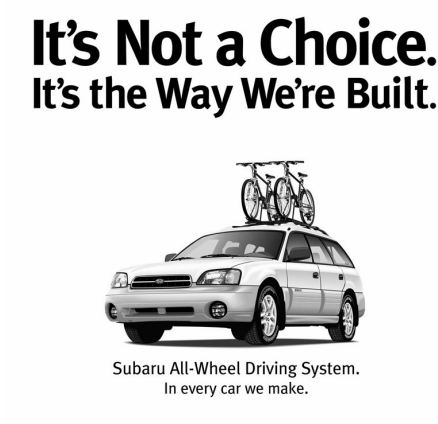
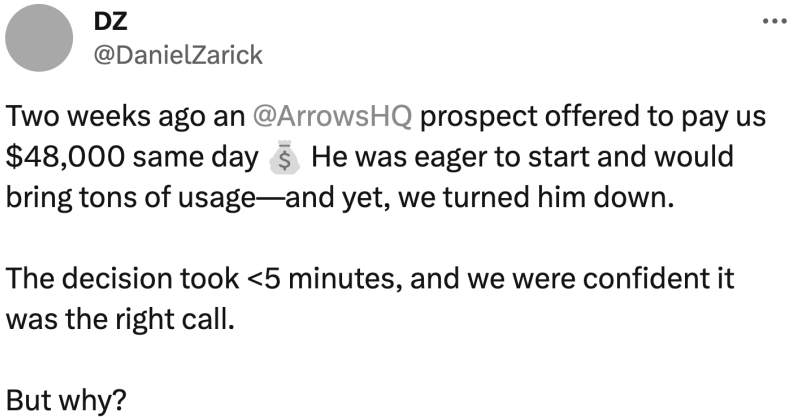


Figure 13: Twenty years before Lady Gaga (p. 1433)'s "Born this way."



credit[331]

When you target Carol, your positioning is strongest, and you win where you deserve to win. But far more people will buy because they have similar purchasing preferences.

FINDING CAROL IN THE NOISE

When a customer cancels, we need find out why—what did they think would happen, and how did we fail them? Unless the customer stopped their project or went out of business, cancellation indicates a gap (p. 9) between what we promised them (marketing) and what we delivered (product).

This investigation is most urgent when the customer is Carol. You stipulated she is the perfect customer, but she disagrees. Perhaps the product needs to change, but perhaps your definition of Carol needs to change to better match what the product has turned out to be or what you are uniquely excellent at delivering (p. 543), or some other facet of your strategy (p. 489).

The same could be said Diana, but when it comes to Eddie it's less clear whether this is signal or noise. Eddie is a mismatch for some of our attributes, and might cancel as a result, but that doesn't mean we should change those attributes, if they align with Carol and Diana. If we treat all customers as if they are Carol, and therefore all cancellation data as a vital signal, we lose track of building for Carol and we end up building for nobody.

Some customers aren't even Eddie—they're not on the bullseye, yet they signed up. You might think this would be rare, since it is irrational, but in fact it will happen all the time. You need to identify these customers and completely ignore their data—the noise will push you off course.

Therefore, you need a way of identifying who is a Carol. Ideally, right from the start, during on-boarding, maybe with a few questions, maybe based on activity indicating they're in that perfect zone. Then, you can measure cancellation rates of Carol (the only rate you care about), and spend extra time finding out why they cancel. The "extra time" comes from the time you saved by not pursuing and interviewing other customers.

PRIORITIZING CAROL OVER REVENUE

Of course it's not just in cancellations that you want to segment the ICPs from the rest of the customers. It's in your activation funnel. It's in your trial or freemium conversion rate. It's in individual feature usage. It's in conversion to monthly active users (meaning a single user actually using it consistently month over month). It's in tech-support requests both for bug fixes and features. It's in every aspect of understanding how customers are intersecting with our product and com-

credit[332]

"All I'm saying is just once I'd like to be sitting without being a target."

pany. It's always about the Carols; everything else is noise, and if you don't separate the two, you can't build the best product, for any definition of "best."

The typical failure mode for this is that your biggest customer, who you desperately want to keep because you really want that revenue, occupies you constantly in tech support and in feature requests, some of which you've implemented just for them, with the internal excuse that "if they want it, other larger customers probably will want it, so this is a path of growth," yet they're still perennially unhappy, unsatisfied, obviously a bad fit, an anti-Carol, but dangit you *need* that money and if we can *just* make this work there *must* be more customers like that coming.

And after all that, of course they leave in a huff, having been incredibly unprofitable despite their high nominal revenue, and you *should* turn around and say, "Wait a minute, that customer was not in our bullseye all along." Rather than noticing that from the start, and helping them to leave your company, you clung to them because you so desperately wanted it to work, and for good reason—it was a lot of money—but ultimately that wasn't the right reason, and the whole thing was an enormous waste of time. You could have been finding

and on-boarding a few more Carols instead, and they would still be here, happily paying you money, profitably.

You'll know that you're doing this right when you turn down a sale, even though it would have been your largest customer, simply because they're not your ICP, as in these miniature stories from Daniel Zarick at ArrowsHQ[333] and John Doherty at EditorNinja.[334] **This separates first-time founders from seasoned founders**. It's not a coincidence that both Daniel and John have seen this movie before, made this mistake before, and are avoiding it this time.

Can you?

After all this, if you reread that ad from SAS with The Power To Know®, you'll die a little inside. So much nothing. Nothing to say, said to no one, caring about nothing, changing nothing, being nothing.

You can do better.

And make more money doing it.

Many thanks to Florian Caeser[335] for contributing insights to early drafts.

Chapter 17:
Product/Market Fit (PMF): Experience & Data

PMF GROWTH · PMF RETENTION

"I had the 'winner, winner' dream again."

credit[336]

Every day people debate how to define "Product/Market Fit."

It's best to start with the subjective experience, because **there is no doubt when you have it**. (If you're not sure whether you have Product/Market Fit, you don't.)

> *If every day it's a struggle to get customers, it's not working yet.*
> *If every day it's a struggle to keep up with demand, it's working.*

It is a momentum change from "push" to "pull"—a change from fighting for each customer, one at a time, to a flood of signups that you can't even explain. It's a change from "how do we get more customers" to "how do we handle the influx of demand?"

It's *working*. Something clicked, like a final puzzle piece snapping into place, a "fit," and suddenly the floodgates are open.

It's not all good. The avalanche of customer complaints outstrip your ability to deliver bug fixes and simple enhancements. You feel behind in every department simultaneously; you can't keep up with the work no matter how "productive" you are. You obviously should hire someone—and this new revenue means you can afford it—but who? Someone just like you, because you can manage them, or someone who complements you, so you can deal with the burgeoning scale? Does it even matter—you don't have time to interview.

It's the proverbial "good problem to have," but it doesn't feel good in the moment, with a crushing workload and unhappy new customers, and your brand slipping away ("I heard they were good but it took them a day to answer my support question" says the public review). Although of course it *is* the *best* problem to have.

You might think all this doesn't matter—you're just running your company; who cares how some article defines a term like "Product/Market Fit?" But it *does* matter, because it determines whether the company is sustainable, and fundamentally changes how you operate each day, how you plan for the future, whether and who you need to hire, and what you need to build, and whether you're going to start tackling scale (p. 773).

(P.S. How do you go from idea to this state? This is my PMF system (p. 9) that I used to build a unicorn.)

What does Product/Market Fit look like numerically?

In my experience building several companies myself and angel-investing in dozens of others (with a wide range of outcomes), **I define Product/Market Fit as *all* of the following:**

1. **Easy growth (pull, not push):** Growth rate suddenly spikes up, and sustains at the new rate. You often don't know why.
2. **High retention:** Cancellation under 3%/mo (for B2B) or 5%/mo (for B2C), because if customers are leaving in droves, you are not a "fit."
3. **Critical mass:** At least $20,000/mo in revenue or increasing WAU's by at least 200/mo. Everyone can grow 1000% month-of-month when the baseline is 7 customers; the fact that you have only 7 customers means there's not yet evidence of "fit."*

What follows are the data to back up this definition, that also matches the lived experience that is the essence of hitting Product/Market Fit: in which life will never be the same again.

* What if you have a single huge customer? Congratulations, lots of people would love to have a huge customer; great start. That demonstrates you have fit for a single customer, but not for a market; that's consulting, not product.

Figure 1: Pallyy's four-year MRR

THE CHARACTERISTIC GROWTH CURVE OF "PRODUCT/MARKET FIT"

Bootstrapped products

When you see growth charts like Pallyy's,[337] you can see the exact moment when Product/Market Fit was reached (Figure 1).

Successful startups often have a growth curve like Pallyy's: Revenue growing linearly, slowly, often for about two years, then in a moment —Product/Market Fit—when it suddenly turns upward, and continues growing at that new trajectory.

The new growth is still linear, but dramatically faster. In particular, it is *not* exponential, though that's the word people frequently use (this universal fallacy is explained here (p. 115)).

ConvertKit[339] was similar, bumping along for two years before growing quickly but linearly, just like Pallyy (Figure 2).

And then that growth continued linearly, not "exponentially" as many pundits like to say (Figure 3).

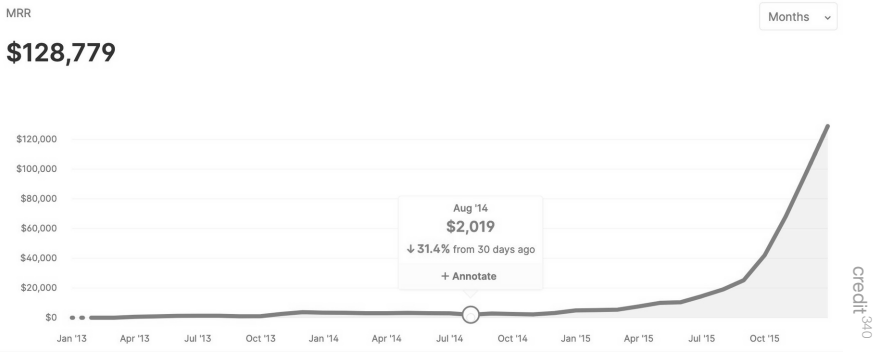


Figure 2: ConvertKit's early MRR growth

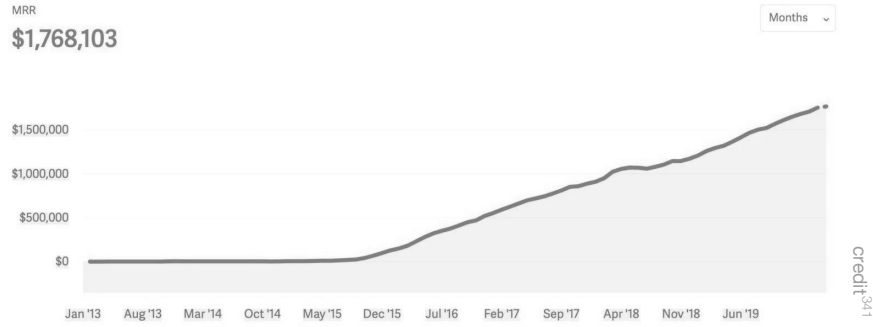


Figure 3: ConvertKit's scale-up MRR growth

You can see the sudden change between "every sale is a struggle" and "the orders just won't stop coming in."

ConvertKit founder Nathan Barry famously tells the story[342] of how he personally reached out one by one to his ideal target customers, manually onboarding them to not only get the sale but to get the testimonial.* It's a brilliant playbook that others would be wise to copy.

It's also what it means to hard-scrabble for customers, and it's really obvious when the forces flipping from having to scratch and claw for every dollar of revenue, to the company running away from him.

VC-funded products

The preceding companies were bootstrapped, but the pattern is the same in the world of venture, because Product/Market Fit is about products and markets and customers and fits, not about funding models. My company WP Engine[344] started 15 years ago self-funded but two years later became VC-funded, eventually raising more than $300M, experiencing hyper-growth** for a decade, yet it exhibits the same two-year slow-growth preamble to high-growth, and both linear (Figure 4).

You can't tell from the chart when we first raised money (it wasn't January of 2012), nor subsequent rounds, nor the size of the rounds. The shape of growth transcends all of it.

I distinctly remember when we flipped to "Product/Market Fit." Only five of us, completely sustainable, and suddenly everything changed. We fell behind on our stellar tech support; one of the dangers of bragging about how "the founder does Zendesk tickets" is that process doesn't scale. We couldn't keep up with our free white-glove site-migration service for new customers. And then, completely behind on all work, trying to stand up new servers and answer phone calls, knowing we needed to hire, we got even *more* behind because hiring takes time—finding candidates, interviewing, not wanting to hire the wrong person just because we're moving fast, and even after we hired,

---

* I was one of them; perhaps you're reading this article because of an email from my newsletter,[343] which is still on ConvertKit.

** Often defined as 2T3D, i.e. in successive years after attaining $1M in annual revenue, the company's revenue triples, triples, doubles, doubles, and doubles, thereby getting close to $100M five years later; WP Engine did that and more, starting with 5x.

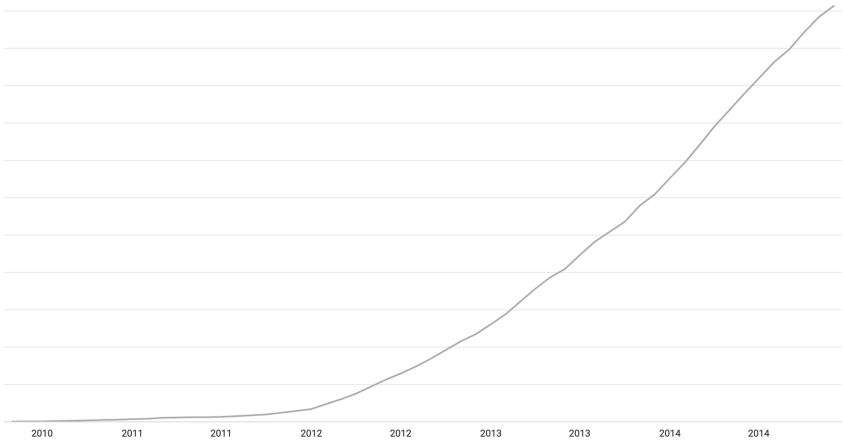


Figure 4: WP Engine's early growth curve

there's no documentation or training, so we're just working side by side, hoping folks learn by osmosis, while the team gets even further behind. And customers were complaining on Twitter that our service is going downhill—and worst of all, they were right. They blamed it on us raising money; what they didn't know is we hadn't spent a dollar of it yet; in fact our error was *not* spending the money on hiring ahead of need, to blunt the impact of a growth increase.

We did get back on top of it later in the year, and we've won all sorts of awards for our service, maintaining 98% CSAT to this day, because that's who we are. But that's not what it felt like in 2012.

The same thing happens to the "cool kids" companies in Y-Combinator (Figure 5).

This isn't a new phenomenon, nor a "SaaS" phenomenon. Peldi Guilizzoni[346] famously shared his (one-time!) revenue for Balsamiq Mockups back in 2008 (Figure 6).

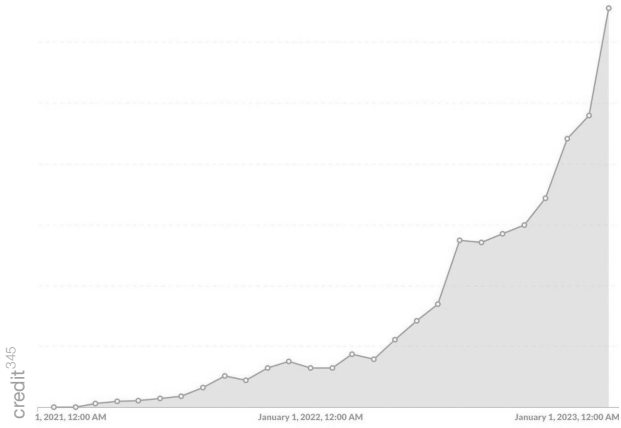


Figure 5: Anonymous YC company growth chart from YC co-founder Paul Graham

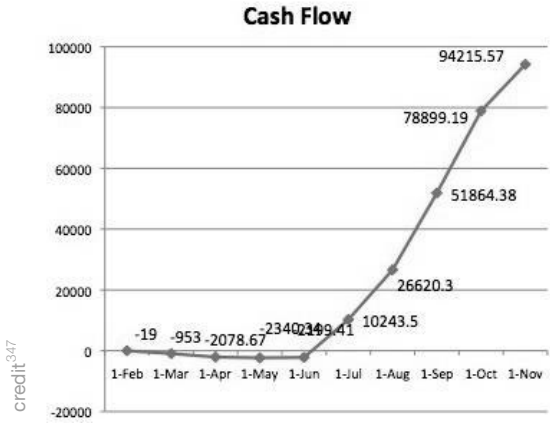


Figure 6: Balsamiq Markups one-time (not SaaS!) revenue, 2008

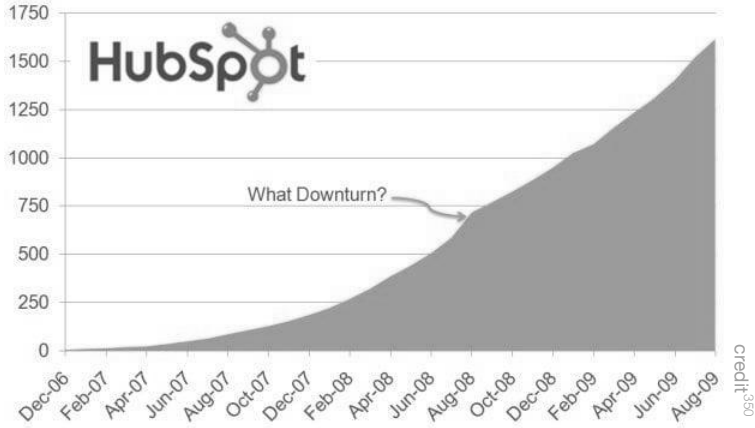


Figure 7: Hubspot's early growth curve

Second products

Hubspot is a hyper-growth VC-funded company, now public. Early on, you can see the linear behavior, fairly consistent even through the 2008 recession, as co-founder Dharmesh Shah[348] indicates on a chart from his personal blog[349] (Figure 7).

Sometimes there's another shift in the slope of the line, when the company adds a second successful product, expands to a lucrative new market, or changes its business model. Hubspot, after twelve years of linear growth, changes to a new (but still linear) growth rate around 2021, around the time their second product-line reached its own fit (Figure 8).

Viral products

All of the previous examples were of B2B companies without "viral" or "network" effects, which are often associated with so-called "exponential growth." However, even classic "viral growth" examples like Slack (at the time the fastest-ever-growing B2B company in history)

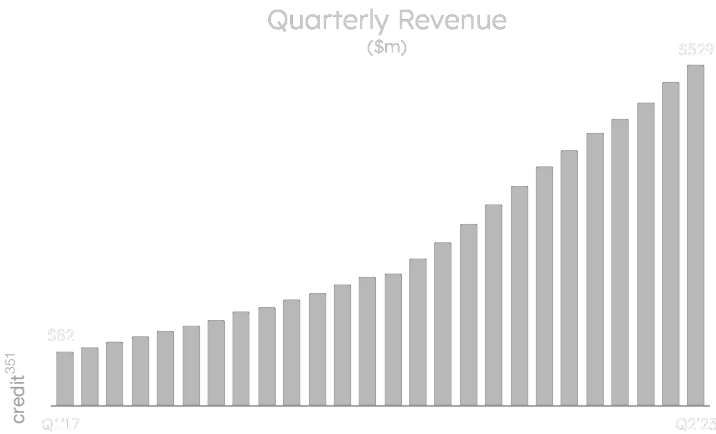


Figure 8: Hubspot's at-scale growth curve, exhibiting a bend, and linear on both sides of the change

and Facebook (at the time the fastest-ever-growing B2C company in history) did not grow exponentially. Instead, they grew in the same way again: linear, then a sudden change, then still linear but faster. Here's Slack, which looks exactly like Hubspot (Figure 9).

And Facebook with a material bend in early 2009 (coinciding with the introduction of the "like" button that you could embed on your own website) (Figure 10).

Elsewhere I've gone into great detail about why even viral products like Slack and Facebook specifically don't grow exponentially (p. 115).

You can also see it in AirBnB with a bend in late 2010, coinciding with the realization that professional photography was the key[354] to "fitting" with customer expectations (Figure 11).

Media products

In many ways, media and content is unlike building and selling products. But in growth curves, and this characteristic PMF curve in par-

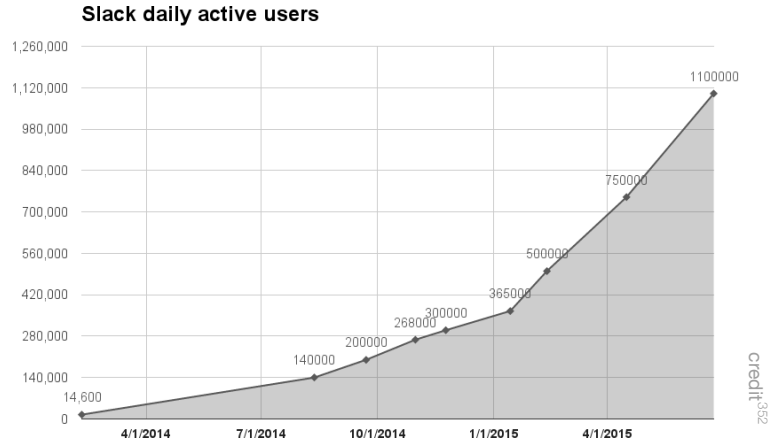


Figure 9: Slack's early user growth curve

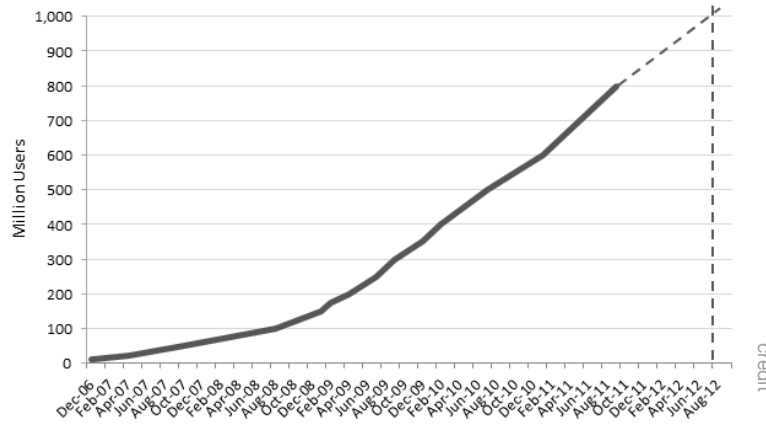


Figure 10: Facebook's early user growth curve

ticular, they are similar. Here's one for a successful YouTube channel (Figure 12).

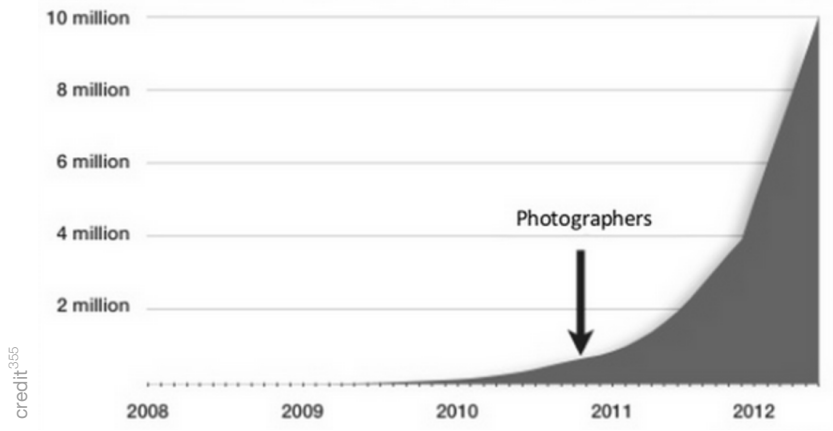


Figure 11: Airbnb's final piece of the puzzle

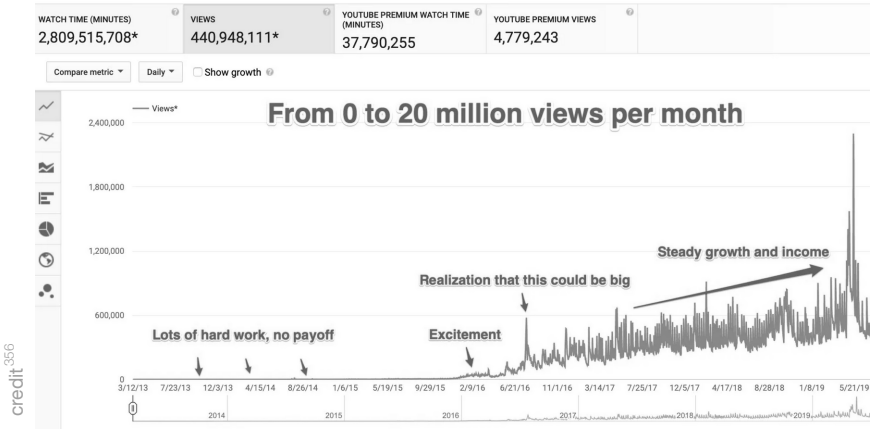


Figure 12

Or email newsletters like Lenny Rachitsky's[357] (Figure 13).

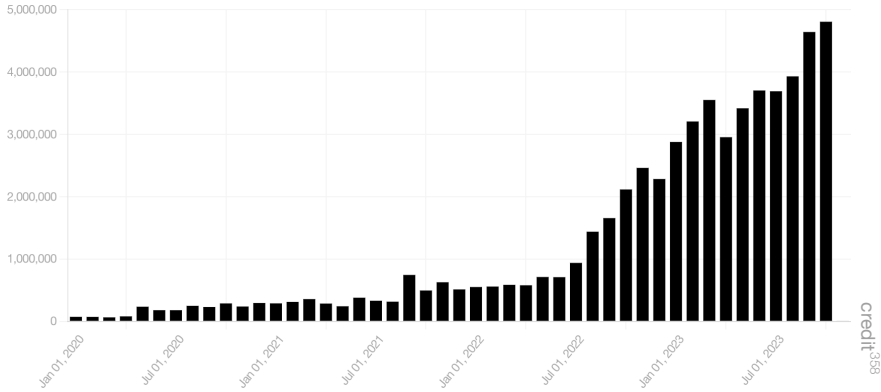


Figure 13: Lenny's Newsletter traffic

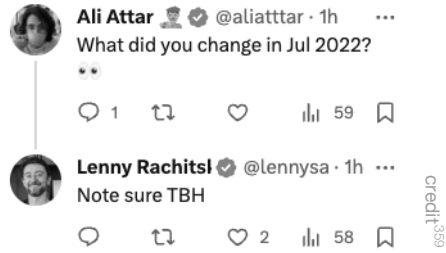


Figure 14

Oh, and what did Lenny do to cause that sudden growth? You guessed it (Figure 14).

Update: Sept 2024: A year later, Lenny published his subscriber chart, and sure enough the pattern continues (Figure 15).

The bottom line is, these growth curves matching "fit" are common with all business models—B2B and B2C, recurring-revenue or one-time revenue, products or media, self-funded or VC-funded.

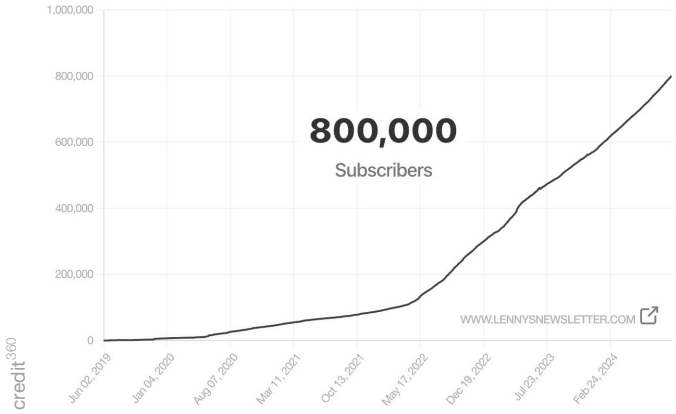


Figure 15: The first small bend in 2020 resulting in a line with a new slope, then the dramatic bend in May 2022 with nearly perfect linear growth for the next two years.

HIGH CANCELLATION MEANS THERE'S NO FIT

There is one killer metric that will stop growth in its tracks, even with the curves above: Cancellation.

Founders often think cancellation of 5-7% per month is OK so long as they're getting lots of new customers. But they are mistaken.

High cancellation puts a ceiling on growth. This is on public display with Buffer,[361] who started with the same sort of linear-change-linear growth curve (2012-2014, then 2014-2020), but then growth leveled off (2020) and never returned (and even shrunk) (Figure 16).

The reason is churn. Note how churn was higher during the slow-growth period of 2012-2014, then reduced to 5%/month, permitting faster growth (Figure 17).

But 5% is still too high; there's no way to keep increasing new-customer acquisition to outpace churn, for the simple reason that

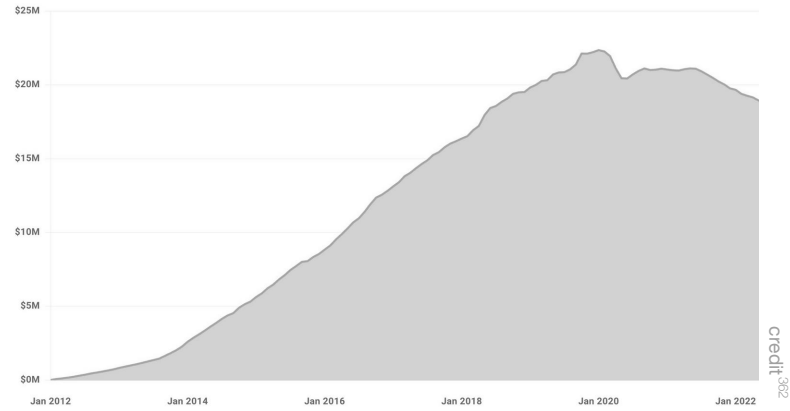


Figure 16: Buffer's ARR

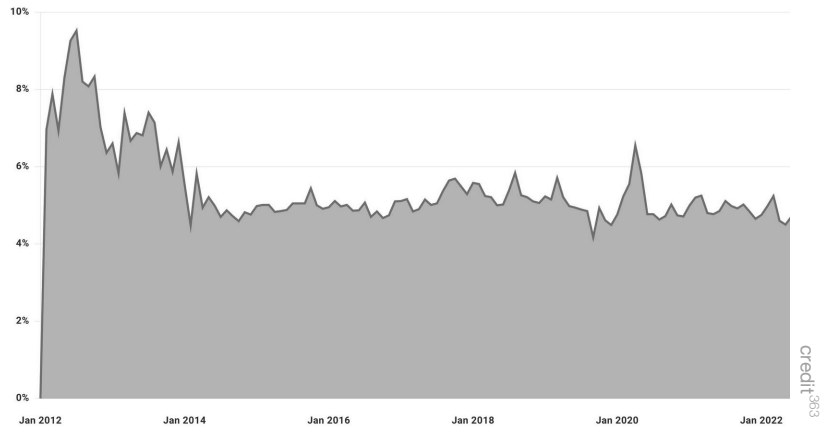


Figure 17: Buffer's monthly cancellation rate

"5%/mo" definitionally grows in lock-step with your total size, but marketing and sales does not continue to automatically grow just be-

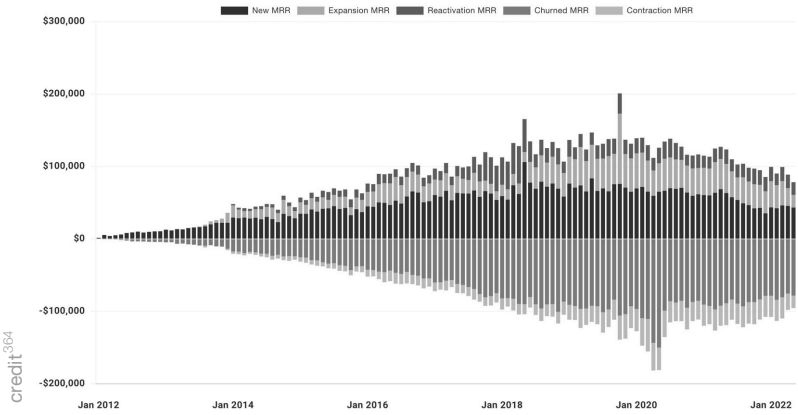


Figure 18: Buffer's unit economics; new-MRR naturally tops out, whereas cancellation never stops growing in absolute dollars.

cause your company is bigger. So, cancellations always win that race, and at 5%/mo, you top out pretty quickly (Figure 18).

You can see a similar effect at SparkToro,[365] an impressive company that is a testament to founder Rand Fishkin's vision to build a great company instead of a company that attempts to accelerate growth forever; he explains[366] how SparkToro generates enviable profit, with happy employees, happy customers, and happy investors. So this isn't an indictment of the company, but it is a demonstration of the growth-destroying power of high cancellation rates (Figure 19).

It's not actually about finances; it's the fact that customers are choosing to leave. The numbers are the objective measure that this is happening, but they're not the point. The point is that customers don't want to stay. That means it's not a fit.

Therefore, both because "lots of customers leaving means it's not a fit" and because high cancellations means the growth curve will quickly flatten, **cancellation higher than 3% for B2B or 5% for B2C indicates a lack of Product/Market Fit**.

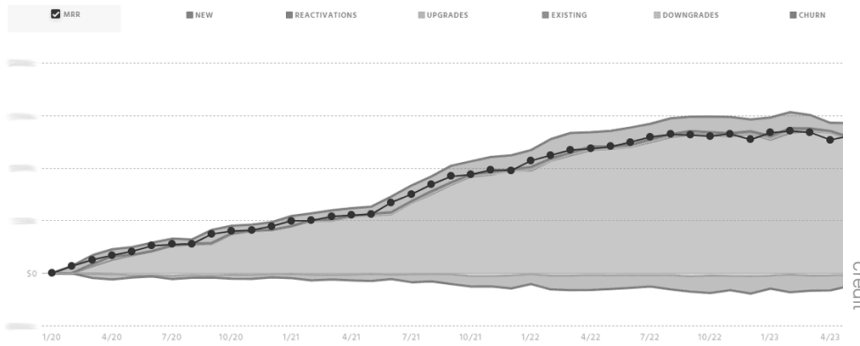


Figure 19: SparkToro is consistent in adding the same dollars-per-month of new-MRR, but their consistent cancellation *rate* continues to *grow* in dollars per month, which means at their current size, cancellation "wins" that race, and growth becomes flat.

Easy inbound growth is indeed a sign of fit, but if customers leave in droves, it means the promise was right, the price was right, but the product didn't deliver, or the customer's need was too brief to sustain a recurring-revenue business.

Some people call this "a fit, but not sustainable" or "a fit, just with limited upside." I call it a lack of fit, because if customers are leaving, it's not a fit.* Don't get distracted by dollars. Pay attention to customer behavior.

If your company is growing slowly, that doesn't mean you failed (p. 1261). It just means you haven't hit that success curve yet, and haven't found Product/Market Fit.

Maybe you never will, but you'll grind out a profitable company anyway. Maybe you never will, and growth is so slow, it's best to face facts (p. 657), stop, figure out why it was too hard (p. 71), and try

* See this amazing five-minute segment[368] from Twitch founder Michael Seibel, describing how his Socialcam app got 60M downloads in 4 months and ranked among the top 5 on the App Store, yet retention was so horrible (almost zero retained users after 10 days) that despite that inconceivable growth, it was not Product/Market Fit.

something else. Or maybe your growth spurt will start today. It's impossible to know (p. 429).

Maybe if you follow my roadmap to Product/Market Fit (p. 9), which I used to build a unicorn, you will succeed. Then again, in that same article I explain that my previous startup used only half of those techniques.

No one knows; that's life. But now you know what it looks like, when it is truly a fit.

Or that you still have work to do, if you're not there yet.

Chapter 18:
How annual pre-pay creates an infinite marketing budget

COST OF MRR · REVENUE WE CAN SPEND
PAYBACK · OPTIMIZATIONS · ANNUAL BILLING
FAQ

"Something's clicking. I want you to find out what, and click the holy hell out of it."

Multiple founders have told me the ideas in this article were responsible for the financial success of their startup.

They might be exaggerating out of kindness, but if it's even 10% as useful for you, it will have been worth your time.

We'll explore how growth affects cash-flow, and conclude with several techniques that can transform the cash-flow of your business.

THE COST OF A DOLLAR OF MRR

What does it cost a SaaS company to add $1 of new monthly recurring revenue?

Using the typical acronyms:

> CAC (**C**ost to **A**cquire a **C**ustomer) is the total cost to get one new paying customer—Marketing and Sales costs, including fully-loaded salaries.* The simplest way to compute it is "total spend in a month" divided by "total new customers added during that month."
>
> ARPC (**A**verage **R**evenue **p**er **Customer**) is the average monthly-recurring revenue you get from a customer. The simplest way to compute it in aggregate is "total recurring-revenue in a month (MRR)" divided by "total number of paying customers during that month (N)".

* In smaller companies, typically the founder and others are spending time on marketing and sales activities as well; include the fraction of their time, or the salary that you would have to pay someone else to do those same tasks. Also include commissions.

Since it costs CAC dollars to get ARPC new dollars of recurring-revenue:

The cost to create one more dollar of MRR = CAC / ARPC

If you haven't done it before, computing this metric will be eye-opening.

Let's posit a hypothetical company with a $10/mo consumer-targeted SaaS product, where they pay $1.60/click for Google Ads; that traffic converts at 5% to a free trial, and those trials convert at 40% to a paid customer. Their CAC is $80,* and their ARPC is $10, therefore we would say "it costs them $8 to create $1 of MRR."

It's tempting to conclude that "It takes 8 months of customer revenue to 'pay us back' for the marketing and sales costs of getting that customer. That customer is unprofitable before then, and becomes more and more profitable after." (Figure 1)

But, unfortunately, it's worse than that.

* $1.60 / 0.05 / 0.4 = $80

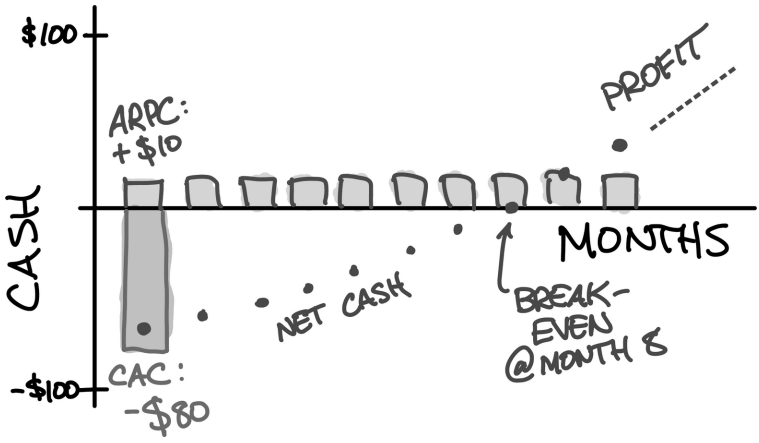


Figure 1

REVENUE WE CAN SPEND

All companies have mandatory expenses associated with delivering the entire product to the customer. Not just the product itself, but everything else the customer expects, like being able to pay with a credit card or call tech support:

- credit card processing fees
- tech support*

* Even if you're a solopreneur, doing support yourself, wanting to claim that therefore "support costs me nothing," the opposite is true: Your time is valuable; you could have used that time for anything else, such as marketing or building a new feature. How should you account for this? Use whatever it would have cost to hire someone else to do the service for you, and remember that low-wage people who aren't fluent in your language, can't provide the level of service you're currently providing!

- infrastructure (if SaaS)
- professional services (if consulting)
- bill of materials (if physical goods)
- *Anything else which, if missing, the customer would say "You're not delivering the product I paid for."*

In finance we measure these costs relative to revenue:

`GPM` (**G**ross **P**rofit **M**argin) is the percentage of revenue remaining, after subtracting these "expenses required to deliver the whole product and experience."

Continuing our example, suppose credit card fees are 3% of revenue, infrastructure costs are 5%, and tech support is 25%. Then `GPM = 67%`, i.e. after fulfilling our promises to our customers, we're left with 67% of the money they gave us.

PAYBACK PERIOD (P)

Now we're prepared to compute the *actual* pay-back period: The amount of time needed to recover `CAC` expenses, which is the number of months of revenue *we can spend*:

`p` = "payback period" = `CAC / (MRR × GPM)`

In our example, `p = 80 / (10 × 0.67) = 12` months. (Figure 2)

The cash-flow implications are harsh. Consider:

You're bootstrapped. You scrape together $1000 from life savings or credit card debt. You spend it on ads. Happily, it works! With the

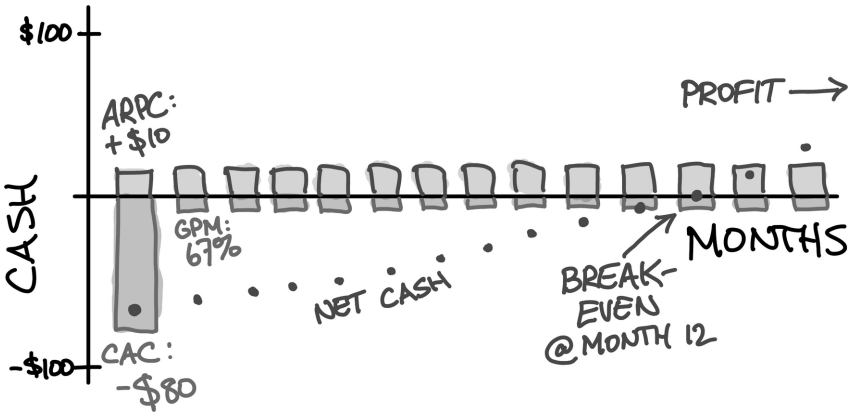


Figure 2

numbers in our example, you now have $125 in MRR, but expenses are $42/mo, so it's going to be a whole year before you get that $1000 back. Meanwhile, you're just *out* that cash.

So... how do you get more customers? Another $1000... but how many times can you do that before you run out of savings? Now you can see why people raise money...

And **the faster you grow the more cash it takes**, because no matter how much you spend, it still takes a year to get it back. Now you see why people raise money for companies that are designed to grow quickly, and why it always takes more money than you think.[370]

A FEW OPTIMIZATIONS

Your mind immediately jumps to the question: How can I improve the metrics in this equation, so that my pay-back period is reduced, so I can grow more quickly with less cash?

Reduce CAC

Of course you'll optimize advertising.[371] However, everyone does this, and most advertising is an auction model, which means it's difficult to find spaces that simultaneously (1) get enough traffic to matter and (2) are low-cost. It's still worthwhile to optimize, or to try to find less-expensive channels, but this is incremental; it won't transform your business. In fact, CAC *increases* as a company scales (because low-hanging fruit is already plucked) and *increases* as a market matures (because of auctions from more and more competitors, who have more and more budget), as I've documented (p. 115).

Some people scream "Social media is free! SEO is free! Content marketing is free!" It's not free—you have to write, manage, jockey for attention, follow Google's whims, Twitter's whims, YouTube's whims, Instagram's whims, TikTok's whims, even though they don't tell you what the whims are, and the whims change, optimize site layout as well as content, post often "for the algorithm," and so on. These might indeed be great channels for you, but let's not pretend that they are "free," any more than you are "profitable on day one (p. 371)."

Increase MRR

This is one of the main reasons why so many startup pundits (including myself) are always telling you to "raise your price." The difference between $39/mo and $19/mo might not matter to your customers,* but can cut your payback period to 1/3 of what it currently is.**

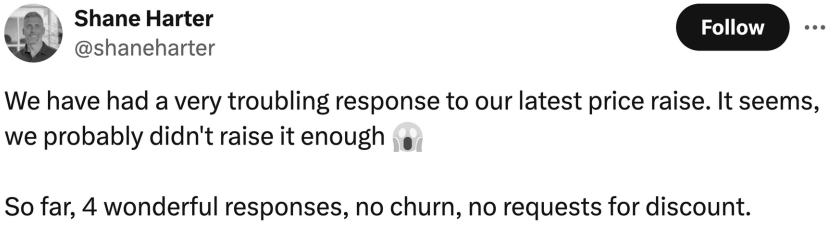
---

* I know it sounds like doubling your price would dramatically reduce how many people sign up. However: (a) If sign-ups halve, you still have the same-sized business, except far more profit, and higher-quality customers, and (b) my experience and many others' is that signups don't halve; they might even go up. Examples below.

** Cut into a third, not a half, because MRR doubles but the costs behind GPM do not, which means GPM also increases, so you get double-benefit.

credit[372]

"I miss the old days when we could just tout things."



credit[373]

You're also probably not positioning the product properly. Here's how you can 8x the price (p. 165) through better positioning, or how you can increase willingness-to-pay (p. 275), which then allows you to increase price and growth and get word-of-mouth (which means lower CAC!).

You probably need more encouragement to summon the courage to try higher prices, so here's a bunch.

Bhanu Teja P
@pbteja1998

Follow

I raised prices for feather.so almost every month now.

This is how it all started...

...me freaking out when I raised it for the first time and @damengchen saying that everything will be alright!

Raising prices is one of the best things that I have done!

credit[374]

Ayman Al-Abdullah
@aymanalabdul

Follow

Friendly reminder that you're not raising your prices enough

Just got hit with this $10 price raise email

Immediately deleted and moved on because I love this service

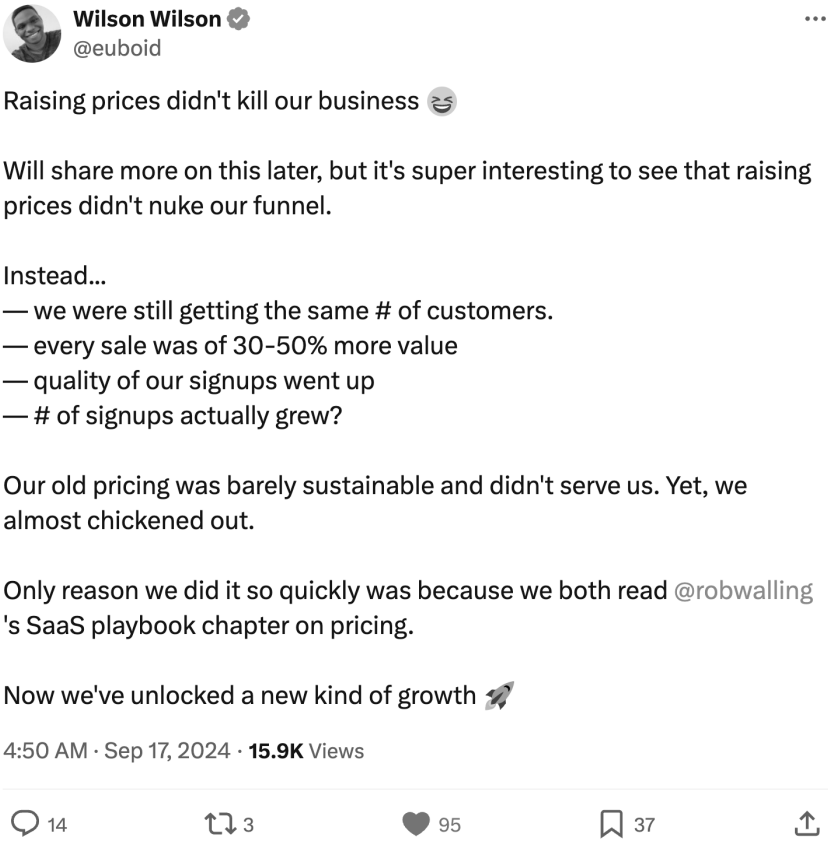
This is immaterial to me but will likely double this companies profits

credit[375]

It doesn't always work, but typically, startups are under-charging.

Increase GPM (by cutting costs)

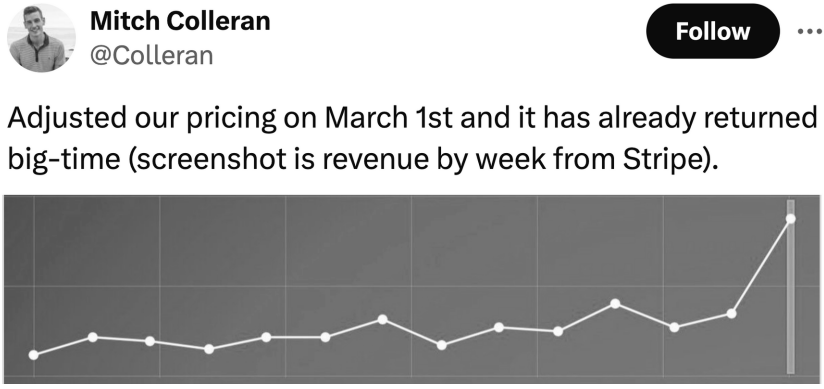
This is tempting, because it's easier to cut costs than to raise prices, and easier to automate some tech support tickets or infrastructure rather



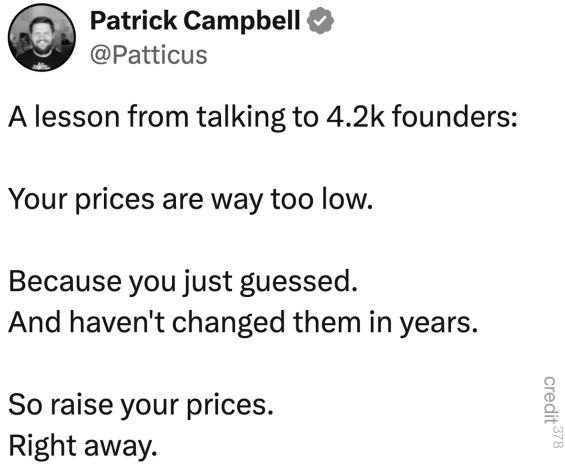
credit[376]

than finding a new advertising channel. Especially for Engineers, who constantly fall into the trap of doing engineering stuff (p. 635) rather than what the business actually needs done (p. 1541).

Of course this *is* valuable work, and if it's easy to cut support-time in half with a few days of keyboard macros or better documentation or an AI first-line-of-defense, then do it. Just be warned that your natural tendency will be to file the rough edges off of costs—which isn't impactful—rather than work on pricing and marketing—which have the power to transform the trajectory of the company.



credit[377]



credit[378]

Remember, you can only cut costs so much, but the potential upside of better marketing or pricing is uncapped. This is one of several reasons why better marketing (p. 259) and pricing (p. 515) is strategic (p. 489), while cutting costs isn't.

THE TRANSFORMATION: ANNUAL BILLING

There's another thing we can try: Billing plans where the customer pays annually instead of monthly.

How much should they pay? Typically you reward the customer for their up-front payment with a discount. In our example, the product is $10/mo; you might offer a plan at $100/year, exclaiming "get two months free!" on the pricing page.

This is a great excuse to raise prices slightly: Offer the annual plan at 12 times the *current* monthly price, then increase the month-to-month price by something like 20% (i.e. the "2 months free" amount). Existing customers can be grandfathered in, or ask them to switch to annual to preserve their monthly rate.

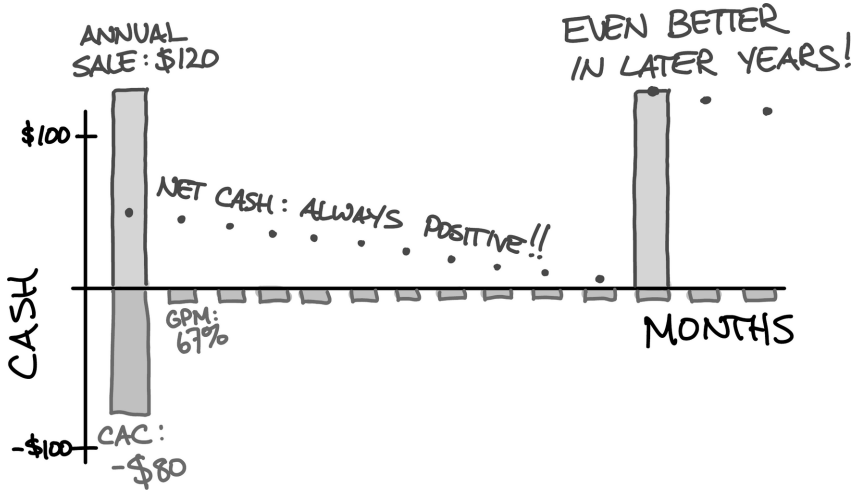
What does this do to our payback period?

In terms of cash:

- Before: We paid $80 to get $10 of MRR, which is $120/year.
- Now: We paid $80 and also got $120/year, **but we got the $120 today.**

This gives us an infinite marketing budget. To see why:

- We spend $80 on ads. But *not* in cash; it's on our credit card.
- We get $120 in our bank account, *in cash*. We mentally set aside the GPM costs that we will incur, that this income will need to



cover. Those costs are $40, which leaves us with $80 that we're free to spend.

- *Later*, our credit card bill comes due, and finally the $80 ad cost leaves our bank account.
- **Which means we never actually lost the $80, in terms of cash!** We received the $80 from the customer *before* we sent away the $80 for the ad, even after setting aside GPM costs.

This completely transforms the growth potential of the business, without raising money.

In real life, not all customers select annual plans. However, this is still transformative because:

1. Even if only a third of customers select an annual plan, it's still a huge impact to cash-flow.
2. You could raise prices on the monthly plans even further, which both (a) increases selection of the annual plan and (b) decreases the pay-back period of the monthly plan.
3. You could offer an annual plan only. (However an "even more expensive monthly plan" is typically a better way of saying yes instead of no (p. 621), as it retains optionality for the customer and is very profitable for you.)

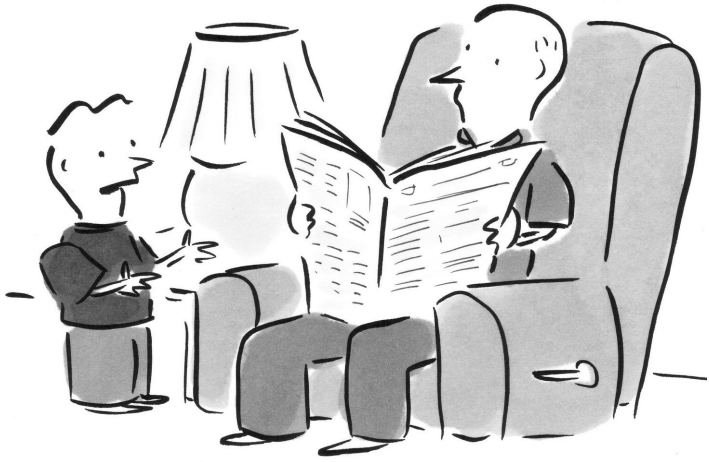
COMMONLY-ASKED QUESTIONS

After espousing this policy for more than a decade, here are the common questions and conversations:

I hear that annual-plan customers cancel less often. So, will encouraging annual plans also decrease my cancellation rate?
No, but it is useful in cancellation analysis.

The reason people on annual plans have a higher retention rate, is that they self-select into those plans. What type of person does that? A person who is already intending to stay. So, offering annual doesn't change your overall cancellation rate, but it does segregate customers in a useful way: Those more likely to cancel are now identifying themselves.

Which means, you can contact them and learn more about why they feel that way, and possibly then do something to improve cancellations, which is likely one of the most valuable things (p. 1191) you can do. You can also compare their behavior and responses with those on annual plans, to tease out what's actually different between the cohorts.



"It's not just a raise in my allowance, it's also a boost for consumer confidence!"

credit[379]

When someone on an annual plan cancels after, say, seven months, should I refund them the remainder of the term?
You can pick anything along this scale of generosity:

Refund pro-rata.
If they cancel after seven months, then refund $\frac{5}{12}$ of their original payment. This has the benefit of simplicity and promoting your reputation as being generous.

Refund as if monthly.
In retrospect, the customer is acting like they are on a monthly plan, so compute what they *would* have paid in that case, and refund them the difference. For example, if the annual plan is $\$100$/year, and the monthly plan is $\$10$/mo, and they canceled after 7 months, then they would have paid $7 \times \$10 = \$70$, therefore you refund them $\$100 - \$70 = \$30$, which is less than if you refunded 5 months of their annual price, i.e. $\frac{5}{12} \times \$100 = \$42$. This also means if they cancel very late in the year you don't refund them at all, as a monthly plan would have been even more expensive.

Credits.
Rather than refunding cash, provide credit on the customer's account. Perhaps they'll buy something else in future, or they want to switch to a less expensive plan, or switch to monthly billing.

No refunds.
We have all experienced this with contracted services like our phone bill or corporate services. This is of course the least-generous approach, so you have to decide whether this policy adheres to your cultural values (p. 827), your sense of business ethics, and whether you want to employ this tool as a type of coercive relationship (p. 275).

What if the customer cancels before the payback period elapses?
Astute readers will notice that the "payback period" model didn't include cancellations.

Clearly, not all customers who sign up on day 1 will still be here on day 30, much less day 300 or day 3000. In particular, if they cancel before the payback period, that's a net-loss for the business. Don't we need to account for that somehow?

Yes! Cancellations were omitted here for simplicity, but in practice you cannot ignore them. A simple way to account for them, is to first calculate your retention rate across the nominal payback period. For example, if payback is 12 months, what is your 12-month retention (`r`)? Excellent for Enterprise software is `r = 95%`; for SMB is `75%`; Shopify is just `50%`.

Then, we declare that the customers who *stay*, must make up the costs for the customers who *leave*. Therefore if the nominal payback period is `p` (as defined earlier), we actually need to wait `p / r` to get paid back.

This formula is not exactly right either, but it's a simple method that gets you to the right vicinity.

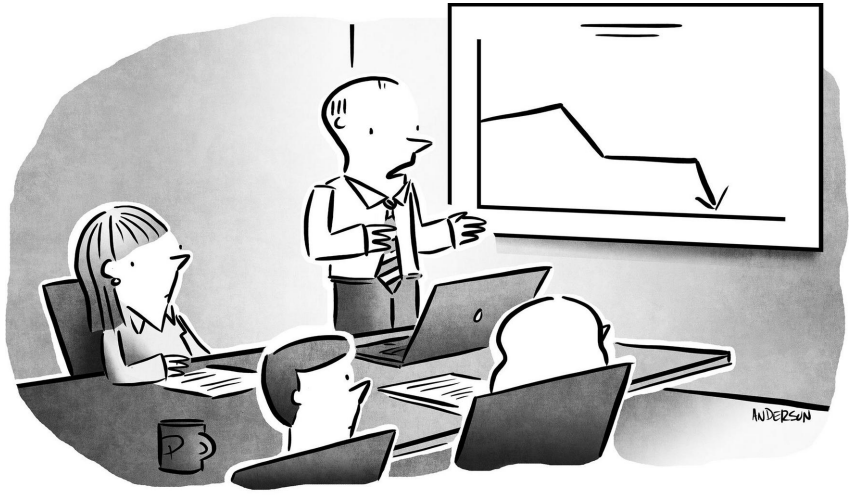
What about all the other expenses at the company? CAC isn't the only cost.
That's right, saying "infinite budget" is artistic license to make the point, which is that it's transformative for cash flow.

I hope you can use annual plans to dramatically change your cash-flow!

Many thanks to Fabio Caravita[380] *for the idea of providing credits instead of cash refunds.*

Chapter 19:
Profitable on day one!

YET! · DON'T DESPAIR



"If by profits you mean an excess of revenues over outlays and expenses in a business enterprise over a given period of time, then yes, it doesn't look good."

> *"My company is profitable, and has been from day one."*
>
> *—every high-tech bootstrapped founder*

I know what you *really* mean.

What you *really* mean is that the only charges on your Business Select PayPal MasterCard this month are:

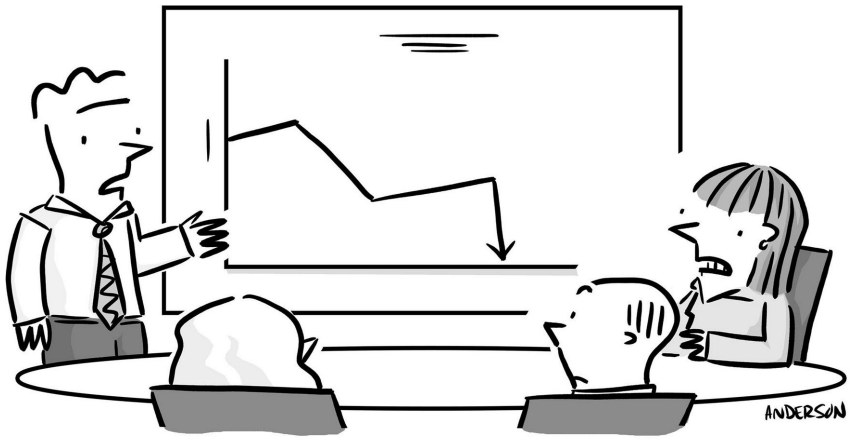
- AWS "medium" instance for the SaaS with one paying customer: $40
- Stripe fees for the one paying customer: $1.72
- AWS S3 storage to back up your loose "Best of Pat Benatar" mp3s: $0.043

And since your one paying customer is at the $49/mo tier, you're **profitable**!

I know that's what you *mean*, but when you say "I'm profitable" it's a turnoff, because it's actually bullshit, and anyone with a modicum of experience knows it. Which means either you're coming off as a full-blown bullshit artiste or—more likely, since we're giving you the benefit of the doubt—you're coming off as ignorant. Neither is a good look.

Let's be serious about what "profitable" means:

- If your savings is going down every month, you're not profitable yet.
- If you're making less than minimum wage, you're not profitable yet.



credit[382]

"OK, let's not get into profit shaming here."

- If you couldn't afford to pay someone else to do the job that you are doing, you're not profitable yet.
- If someone bought your company, and hired people to do the work (even at below-market rates), and each month they'd have less money in the bank than the previous month, you're not profitable yet.
- If you have to work another job to pay the bills, you're not profitable yet.

YET!

The key word is "yet." You're not a failure (p. 1261). You're not doing anything wrong. It's just that businesses aren't profitable on day one. They're profitable later (if ever).

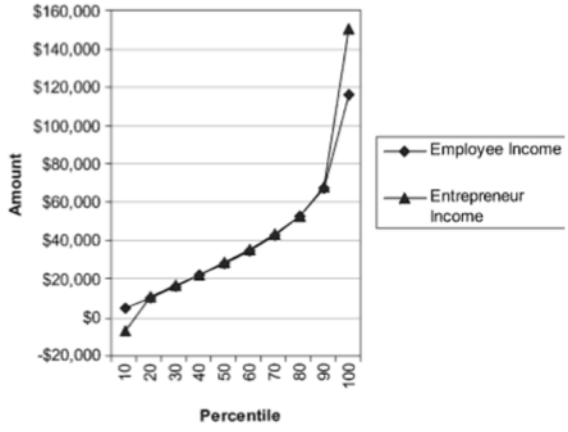
The main error is ignoring the cost of yourself. Although arguably your time is worth $1000/hr (p. 1413), let's just say you need to pay yourself "enough." Which means what?

An easy definition is: You are *taking out of the business* the same amount of money you were making at the job you quit. (Oh, you haven't quit yet, because you need the money?) The market has already decided your time and skills are worth at least that; if the startup replaces it, it's fair (in my book) to call it profitable.

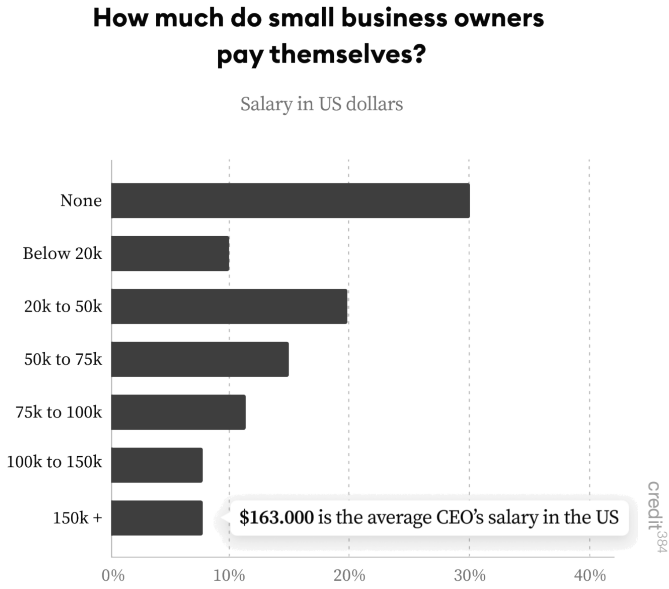
Indeed, the majority of small business owners make about the same money as they were making when they were an employee, without all the stress and extra hours:

And most small business owners make less than the average CEO salary:

At minimum, you need to be ramen-profitable,[385] i.e. enough for you to eat as cheaply as possible and not be malnourished, and also not be burning savings. You can dramatically change that income requirement through lifestyle and location; this is why it can be easier to start a company when you're young and unattached and childless, when you have few material needs and might like living in another country for a while.



credit[383]



The strongest definition of profitability is to imagine that you've sold your business to a high-quality operator who is hiring people to do the work. You likely "wear a lot of hats," so some of those roles are part-time. You might be excellent at some—the new owner will have to hire great talent at market rates—and poor at others—the new owner can skimp on those. The company is profitable when it would also be profitable in that scenario. In short, when *it* is profitable on its own power, not "it + a founder pouring their heart and soul into every waking hour to the limit of their endurance and ability, for free."

DON'T DESPAIR

So now that I've perhaps unfairly ridiculed you, let's take a step back and recognize what's *really* going on, because it's wonderful and amazing and fantastic and exciting:

You're building a business! Sure it's just begun, sure it might need a kick in the ass, sure it might be struggling, sure sure sure, so what? You and every other little new business. You and everyone else who doesn't explode out of the blocks. Almost no company explodes out of the blocks, including all the successful ones in all industries (p. 335). This is exactly what you'd *expect* it would do, even if you're actually the next 37signals or Smart Bear or WP Engine.

Indeed, that's exactly what my company WP Engine looked like for the first 9 months. And then it was chaos for two years, like it always is (p. 429).

Same with my previous company Smart Bear—it took 2.5 years before I could even hire one employee, and even then it was 1/4 of the salary he deserved (and later ended up making). Eventually we, too, made millions of dollars a year—in profit!—but not for years.

In other words, there's nothing strange or bad here. It's just that it's not "profitable from day one." Stop saying that.

Dispense with the feather-fluffing and get to what *is*—the strengths you have, the challenges you want to overcome, the resources at your disposal.

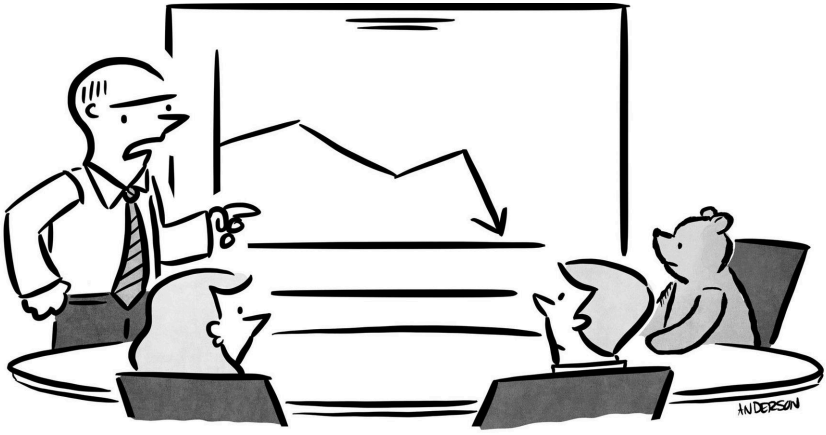
And then set your mind and goals on making that sucker profitable *for real!*

credit[386]

"And we all lived profitably after."

Chapter 20:
Avoid blundering: 80% of a winning strategy

NOT BLUNDERING · BLUNDERING IN STARTUPS
PREVENTABLE BLUNDERS · SOURCES



credit387

"'Oh bother?!' 'Oh *bother*?!' Buddy, we're *way* past '*oh bother!*'"

You have been told:

1. Establish a vision.
2. Set goals against the vision.
3. Create actions that achieve the goals.
4. Measure progress against the goals.

I've told you that myself. It's how you should plan work (p. 1065).

But... what if success is just as much about what *not* to do, as what *to* do? What if intentional goals aren't the only way to succeed?

What if 80% of winning comes from: Not blundering.

NOT BLUNDERING YOUR WAY TO VICTORY

Chess Grandmaster and chess-world-famous teacher / coach Ben Finegold has a simple explanation for who wins amateur chess games:

> What most people say is: "That guy is better than that guy, so that guy won."
>
> But that's *not* why that guy won. Normally the game is very close, and then someone blunders and now it's over.
>
> If you look at it with a computer, it will say "no one's winning" and then "white's winning" and then "black's winning" and then "no one's winning again" and then someone blunders and they lose.
>
> —Ben Finegold, YouTube[388] *(with light editing)*

Not strategy, not memorizing opening lines, not practicing your end-game technique, not studying the Great Games of History, not drilling with puzzles to get better at tactics, ... just *blundering*. Yes, *good* chess players need to do all that stuff, but we scrubs just need to "not blunder."

Is this true for me? I'm a decidedly mediocre chess player, despite countless hours of videos, puzzles, courses, and playing. Are the outcomes of my games dictated by blunders, like Ben says?

Fortunately, I have perfect data. I have hundreds of games with people who are of similar strength.* Chess.com creates an analysis of every game, including the number of blunders, mistakes, and inaccuracies** made by each player (Figure 1).

I checked my last twenty decisive games. I won exactly 50%, which is further evidence that the players were of equal skill, and that the analysis includes both wins and losses.

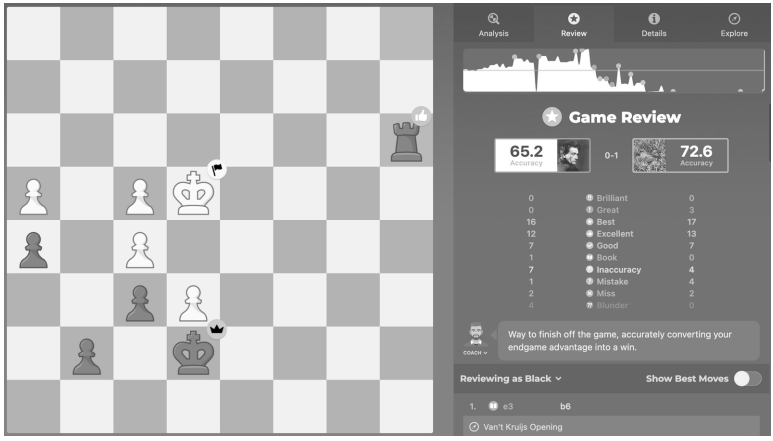


Figure 1: I won, thanks to four blunders by my opponent. We both suck at chess. As Ben says: The truth hurts.

* As evidenced by an even win/loss record and a similar rating.

** The definition of these terms is not important here; suffice to say they're in decreasing order of "how terrible that move was," and the analysis is consistent and sensible across all games.

Theoretically, the best estimator of "who will win" is whichever player has a higher Elo rating.* In these games, the player with the higher Elo won 55% of the time—barely more predictive than a coin flip. This makes sense, because the players were close in Elo,** which means the players are of equivalent strength, so each ought to win half the time.

In my games, the player who committed more blunders lost 86% of the time. Ben is right!

However, in 40% of the games both players had an equal number of blunders. So I also included "mistakes"—the next-worst kind of error. I computed a simple "error score" that includes mistakes while giving blunders more weight:

```
[number of mistakes] +
2 × [number of blunders]
```

Now almost all the games received a decisive prediction, and the player with the higher error score lost 81% of the time.

To win at chess, blunder less.

This is true across many sports. In a wonderful article,[390] Shane Parrish recounts Simon Ramo's analysis of amateur*** tennis games, in which he found that 80% of the points were *lost* rather than *won*. Meaning, 80% of the points were awarded to player P because player

---

* Elo rating[389] is an objective rating system used by all chess players for the past fifty years. It shifts after every game, calculated by which player was higher rated, and by how much, and the result of the game: win, loss, or draw. So, a higher-rated player who beats a much lower-rated player results in a minuscule change in Elo, because that was the expected result, whereas a much lower-rated player who just draws a higher-rated player will gain significant Elo.

** Games had an average difference of 74 Elo points out of around 1100.

*** This observation applies only to amateur tennis. In professional tennis it's just the opposite: 80% of the rallies are *won* rather than *lost*, as unforced errors are infrequent. This is true in chess as well, as high-level players don't blunder, and thus it really is that litany of other skills that results in high standings.

Q made an error such as hitting the ball out of bounds, or into the net, or double-faulting. Only 20% of the points were "won," such as a powerful return that landed out of reach of the other player.

Ramo concludes the same thing that I concluded about chess, even invoking the word "blunder":

> ... if you choose to win at tennis—as opposed to having a good time—the strategy for winning is to avoid mistakes. The way to avoid mistakes is to be conservative and keep the ball in play, letting the other player have plenty of room in which to blunder his way to defeat, because, being an amateur, they will play a losing game and not know it.

Could the same rule be true of other complex systems, like startups? My amateur chess games were 80% determined by blundering, and Ramo's amateur tennis points were 80% determined by blundering.

Could "not blundering" be 80% of winning?

BLUNDERING IN STARTUPS

It's common to say "startups die because they ran out of money" or "startups die because the founders gave up." But that's like saying someone died because their heart stopped, without mentioning that they were battling lung cancer. Doctors call "heart stopped" the "proximate cause"—the last thing that happened—as distinguished from the underlying cause.

We need to identify the cancer, not just say "the founder stopped." Otherwise, the advice for how-not-to-blunder would be "never stop." But "never stop" is not helpful, and indeed not correct (p. 159).

But it's not turtles all the way down.[391] We mustn't dig too deep, as tempting as "5 Why's"[392] might be. Maybe they got cancer because

they smoked a pack of cigarettes every day, so "smoking" is the cause. Maybe they started smoking to fit in with a crowd, but "wanting to fit in with the crowd" was not the cause of death, and it would be incorrect to conclude that "fitting in with the crowd is unhealthy." (Though sloughing off that activity might help you avoid unhappiness (p. 399).)

So we need to identify the blunders which aren't merely proximate, but aren't so distant that they are irrelevant.

There are many sources that claim to finally explain why startups fail. I've provided a raw list at the end of this article. I've deduplicated and rearranged those into the lists below, and added a heavy dose of my own opinions.

Proximate causes

First, let's dispense with the proximate causes, as these are not real reasons, and we should neither repeat nor analyze them:

1. **Ran out of cash.**
   (A consequence of failure, not a cause. Unless you over-spent with no plausible way to recover it with revenue.)
2. **Pivot gone bad.**
   (the reason you pivoted is because it was already failing)
3. **Founder stopped.**
   (As opposed to what?)
4. **Didn't find Product Market Fit (PMF)** **(p. 335)****.**
   (a restatement of "it didn't work")

Bad luck

Perhaps there was nothing you could have done to predict or prevent the failure. That makes it no less real, but perhaps there's nothing to learn from it, or to do differently next time:

1. **Sudden, dramatic shift in the economy.**
   (If COVID bankrupted the movie theater, there's no "strategy" they should have already had in place to prevent it.)
2. **Surprise co-founder break-up.**
   (Some of these are preventable—see below—but if someone falls ill, or breaks a promise, or bails, or commits fraud, it's unclear what should have been done differently.)
3. **Known Brittle Point**
   Brittle Points (p. 905) are single-points-of-failure in the business. All new businesses have them; it takes time and investment to address them. This is not a blunder, but a known risk. Sometimes, "known risks" get triggered.

What *doesn't* kill startups

Equally interesting are things that appear on these lists, but I don't agree that it's a blunder.

1. **Bad idea.**
   (We're repeatedly told that all great ideas start out looking like bad ideas (p. 429), ⋆ *so "having an apparently-bad idea" cannot be a blunder by itself. It's not the initial idea, it's in finding the market and in the execution (p. 9), iterating the idea into a working business.)*
2. **Can't make the product.**
   (While this can happen, typically the founders know how to build things; indeed, building the product while ignoring all other aspects of building the business, is often the problem)
3. **The number of co-founders.**
   (It's easy to find examples of successful startups with one or multiple founders, and equally easy to find failures. It's easy to find failures

⋆ Originally attributed to Steven Spielberg. Attributed to Paul Graham[393] in the context of startups. I've also given many examples (p. 193) along with how to build a strategy around it.

specifically due to having just one founder, or due to or having multiple. Even the great Paul Graham says in a single essay[394] *that a major cause of startup failure is "Single Founder" and later "Fights Between Founders." I personally believe that 3 founders is a warning flag and ≥4 founders is a bright red flag, but it's common for some of those founders to bleed off anyway (which is also why it's a red flag). It's more important that the quantity of founders matches the style and personality of the founders.)*

4. **Location.**
   (This is still a valid reason for startups that intend to scale and become Unicorns, because those startups will need to hire many people, in all departments, with experience in those environments, quickly, and that's much harder to do in a location (including "distributed") where the talent doesn't already exist. This is also true of raising money. However this is becoming less true every year, and it's irrelevant if the startup doesn't have that goal.)

PREVENTABLE BLUNDERS

Self-inflicted blunders are preventable, or at least one can dramatically reduce the risk of occurrence. So, if you don't, that's on you.

Didn't talk to customers (and listen)
If you didn't talk to customers before you started, and especially after you started, that's a blunder. If you talked to them and weren't honest with yourself about what they were saying, that's your faulty analysis. You should have used the Iterative Hypothesis Method (p. 239) or some other reputable framework.

No market need / Bad market
If the market was small and stagnant or shrinking, you already knew that. If there weren't enough people who needed the product, you could

have known that. If they don't agree they have the problem, or that it's worth money to solve, you could have known that. If they don't have the budget or don't accept the price, you could have known that. If they're happy with what they have, you could have known that. The checklist of things that make for a good market (p. 71) is well-known; you don't have an excuse for not finding out before spending six months writing code.

> *"There's just one mistake that kills startups: not making something users want. If you make something users want, you'll probably be fine, whatever else you do or don't do."*
>
> *—Paul Graham*[395]

Did not select a target audience, and clearly communicate the value proposition to them

Speaking to everyone means speaking to no one. Speaking to everyone is a blunder; it's laziness or fear of deciding who this product is for. This determines the language on your home page, in your advertisements, in your sales calls, inside your product. Especially now, with AI, there's no excuse for not distilling what is special and vital about you, into language that your target audience can understand. You must sell to Carol, your ICP (p. 317), which also means knowing exactly who Carol is.

Too many things had to go right

All startups have risks, and have potentially fatally large gaps with a high risk of not being solved. But some have too many gaps—too many things to go right simultaneously. If you have nearly all the requisite skills, if nearly all the objections customers have are addressed, if the market is healthy, if there are many possible distribution channels, if there are many possible niches, if more and more money is being spent in the market, then you should take a risk on the remaining gaps. But if all of those things are uncertain or are gaps that you know you'll need some

credit[396]

"Serendipity is up, fluke is doing well, but I'm a little concerned about our dumb luck."

luck to overcome, then the probability that *all* of those things will break your way is near-zero, and you already knew that when you started. Your path should consist of "or's," not "and's." (p. 1277), so you can satisfy the Startup Drake Equation (p. 667).

Founders / investors broke up
This can be bad luck, but you do pick your co-founders, and either you've known them for a while (in which case you made a poor selection) or you didn't (in which case you made a rash decision). One reason why "break up" kills startups is that the departing founder left holding a huge amount of stock; the reason for *that* is there wasn't a vesting schedule for all founders. That is a blunder. In any case, the team did it to itself—it wasn't a competitor, or the economy, or the customers.

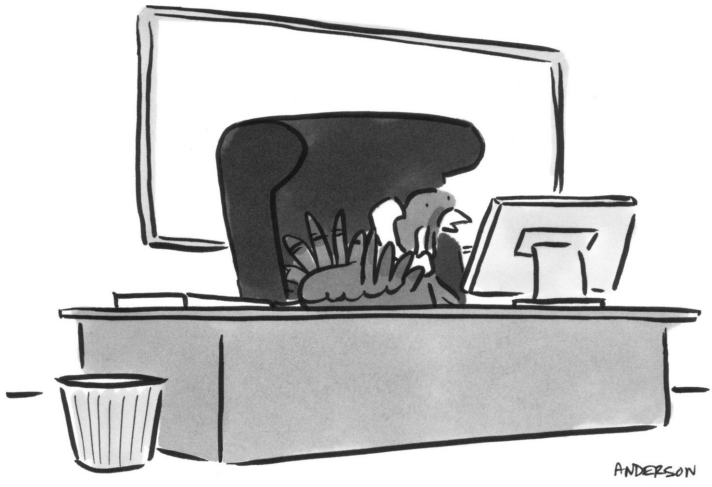
Had no differentiation in the market
This blunder is obvious from day one, when you had no idea why you were special, why the product is special, why someone should buy from you instead of someone else, why you're the right person to build this company, why you will succeed in a space where many others have failed before. There are so many kinds of leverage (p. 543) you could have used, and you chose none of them.

> "Tolstoy opens Anna Karenina by observing: 'All happy families are alike; each unhappy family is unhappy in its own way.' Business is the opposite. All happy companies are different: each one earns a monopoly by solving a unique problem. All failed companies are the same: they failed to escape competition."
> Peter Thiel, Zero to One[397]

Refused to seek the truth / refused to see the truth / refused to learn
We all have to force ourselves to see the truth, because the truth hurts. The truth is that our ideas weren't right, our insight isn't shared by customers, our awesome design is confusing, our potential customers love that competitor we've been calling "dumb," that thing we say is "broken" is not in fact broken, the pain we insist our customers are experiencing, they're not experiencing. Did you even ask? (p. 239) If you asked, were you listening (p. 845)? If you haven't made major changes to your strategy and product and positioning, then you're blundering. Seek the truth, then face the truth (p. 657).

Launching too early / Launching too late
It's funny that both of these are causes of failure. It's a preventable blunder to launch with a crappy MVP instead of a simple but lovable v1 (p. 101). And because builders love only the building, they build for six months (or two years) without customers. Shipping crap, and never shipping, are both blunders. (P.S. Also, don't "launch." No one cares except your friends, and you're going to have to reliably and consistently get customers for the next three years anyway, so just work on that.)

Premature scaling
Before Product/Market Fit, the job is to discover what works. What customers will actually pay for (p. 275) (and keep paying for a year later), what is the perfect, target customer (p. 317) and what do they want (p. 259), how do you reach them through marketing and close them through sales (whether self-sold or human-sold), what you should charge and how. Many founders either don't know what Product/Market Fit looks like (p. 335), or don't want to know, because they want to declare "I have it!" because it feels good to say and garners hearty congratulations and maybe a touch of envy on Twitter. But scaling is about doubling-down on what works—doing it more, faster, better, higher



credit[398]

"No, the end of November is bad for me."

quality, more efficiently. Spending that time and money before you know what works, means wasting all that time and money. Then you run out of both. A blunder possibly of ignorance, but typically of putting ego before the truth.

Selling to the Enterprise before $20M ARR

If you're targeting the Enterprise segment from day one, and you're already built for the sales, enablement, onboarding, account management, legal, security, and product requirements, then this doesn't apply to you. However, few startups begin there, exactly because of those heavy requirements. The common pattern is to have some success with the small or mid-market, then observe that "Salesforce bought some seats!" and decide to "go upmarket" because "we're seeing signal that they need it." There are always individual teams inside large companies who will buy your stuff—that's good! But that's different from targeting "the enterprise." You probably don't even know what you don't know—a position of weakness that screams "bad strategy." This is an easy blunder to avoid; take the orders from the teams, stick with the leverage you have (p. 1017), and don't confuse a team using a tool on the side with what it takes to make multi-million-dollar top-down sales.

"Listen, just tell us what you know. We don't want you, we want the big one."

credit[399]

Unworkable business model / cannot be profitable
Unit economics can be measured right out of the gate. You already have a rough idea of the costs of production, and you know you'll have to spend money and time on marketing and sales. Mismatching price to the business model (p. 515) by an order of magnitude is preventable. Many companies have no sensible business model even at scale. They say "we'll figure that out later," but that's just admitting you're blundering and hoping your future self will figure out how to stop blundering. Sometimes you will; if you don't, whose fault was that?

Writing code instead of winning customers
After you have a working product in market, why aren't you getting more customers? The builder says: Because I need more features. Or: Because the product has too many bugs. Both are incorrect; there are customers who need only the current features, but you're not working on getting their attention with marketing, piquing their interest with the website, and winning their business with sales. And *potential* customers certainly aren't ignoring you because your product has a bug—they don't even know that yet. Coders like to code; everything else is hard, and not fun, and unclear how to execute, and costs money, and it's so comfy to just sink back into the comfort of Visual Studio and marvel at the prescience of Copilot and tweak a CSS class. So people don't find

you, and don't buy, and the company fails, because you blundered marketing and sales.

Expanding the target market before winning the target market / moving too quickly to the second product
I'm not here to call people out, so I won't actually quote the many times I've seen someone on Twitter declaring that now that they've reached $2,700 in MRR, they are expanding into a new market segment. This simple blunder belies a lack of focus, a lack of conviction in the strategy, and lack of understanding that it is ten times easier to win another customer in your existing target market (p. 317) than it is to expand. This blunder dilutes your attention, marketing dollars, positioning, messaging, and product sharpness, and is easy to avoid. There is a time for expansion (p. 793): Later.

Lack of passion / endurance
It is possible that there was passion at the start, but it evaporated, or there was an honest intention of spending the next ten years working on this mission, but you broke down. But that excuse is only valid if, on day one, you honestly took stock of what your passions were, what strengths you were going to leverage (p. 543), understanding who you are (p. 399), had a mission you deeply cared about (besides "make money" or "be able to say you're a founder"), read about the emotional whiplash (p. 737) that startups always entail, and made the trade-offs against other important things (p. 901) in your life. If you didn't do those things, it's a blunder, not self-discovery. (If it really was self-discovery, you are forgiven, and more importantly, you should forgive yourself.)

In most cases, I've linked to specific ways to avoid each blunder. Which leads to the final point:

There is a way to build startups with lower risk, in large part by avoiding those blunders, which I've summarized in my Roadmap for Product/Market Fit (p. 9).

Is this a fail-safe path? Of course not. Even in chess. The rest of the game does matter.

Still, when you can reduce so much risk by avoiding avoidable things, especially with frameworks that detail what "not blundering" looks like, why wouldn't you do that?

"Arrows are not for everyone. Side effects may include romance, passion, feelings of love, dizziness, headaches, and nausea. Ask your doctor if an arrow in the heart is right for you."

credit[400]

80% winning is not blundering.

ADDENDUM: EXTERNAL SOURCES OF "WHY STARTUPS FAIL"

CB Insights:[401] **From "111 interviews"**
(Figure 2)

Paul Graham:[403] **The 18 mistakes that kill startups**

1. Single founder (as opposed to having a co-founder)
2. Fights Between Founders
3. Bad location

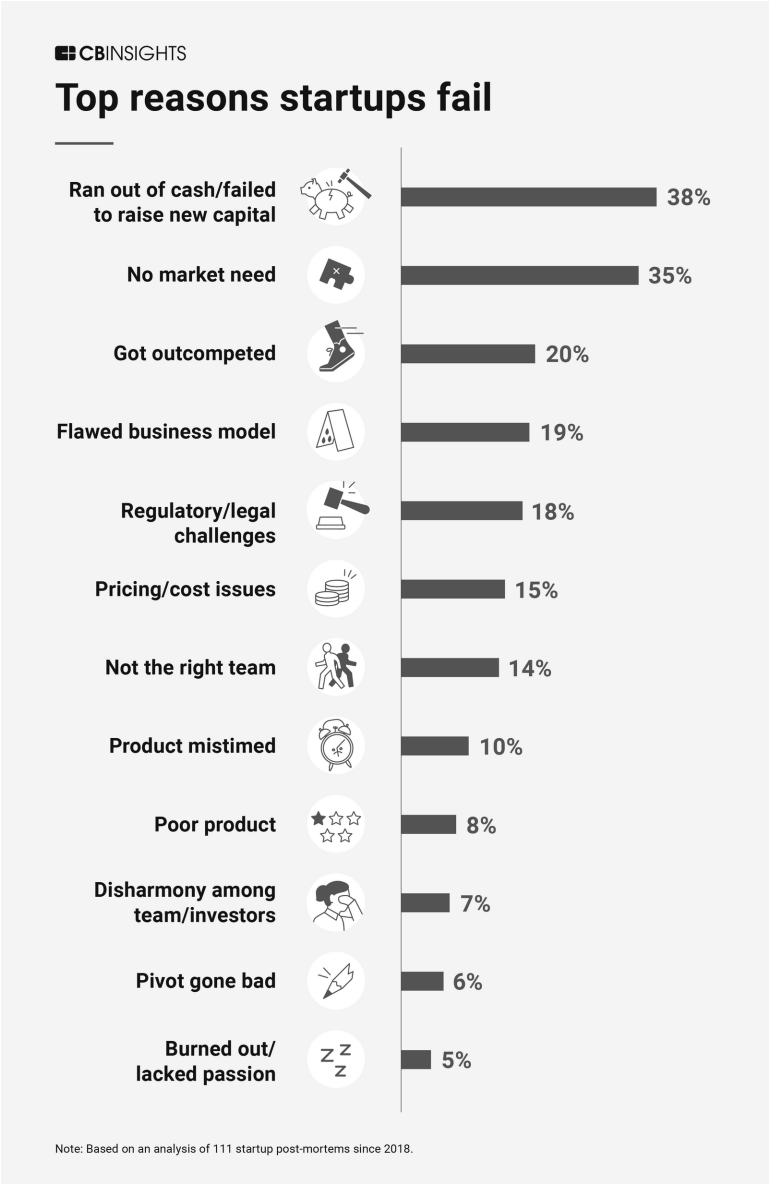

CBINSIGHTS
Top reasons startups fail
Ran out of cash/failed to raise new capital
38%
No market need
35%
Got outcompeted
20%
Flawed business model
19%
Regulatory/legal challenges
18%
Pricing/cost issues
15%
Not the right team
14%
Product mistimed
10%
Poor product
8%
Disharmony among team/investors
7%
Pivot gone bad
6%
Burned out/ lacked passion
5%
Note: Based on an analysis of 111 startup post-mortems since 2018.

Figure 2

4. Marginal niche
5. Derivative idea
6. Obstinacy
7. Hiring Bad Programmers
8. Choosing the Wrong Platform
9. Slowness in Launching
10. Launching Too Early
11. Having No Specific User in Mind
12. Raising Too Little Money
13. Raising Too Much Money
14. Poor Investor Management
15. Spending Too Much
16. Sacrificing Users to (Supposed) Profit
17. Not Wanting to Get Your Hands Dirty
18. A Half-Hearted Effort

Tom Eisenmann:[404] Why startups fail, from hundreds of interviews and surveys

1. Good Idea, Bad Bedfellows (not just founders, but investors, key executives and employees)
2. False Starts (not researching enough before starting or MVP—customers, competitors, market)
3. Maintaining Balance (managing psychology and maintaining a healthy determination)

Steve Blank:[405] 9 Deadliest Startup Sins

(You can see Lean Startup screaming out of these: Answers are outside the building, not in your head, now seek it and iterate.)

1. Assuming you know what the customer wants
2. The "I know what features to build" flaw
3. Focusing on the launch date
4. Emphasizing execution instead of testing, learning, and iteration
5. Writing a business plan that doesn't allow for trial and error
6. Confusing traditional job titles with a startup's needs
7. Executing on a sales and marketing plan
8. Prematurely scaling your company based on a presumption of success
9. Management by crisis, which leads to a death spiral

John Osher:[406] **17 common mistakes from consumer products**
(Many are variants of "scaled too quickly" or "spent money ahead of need.")

1. Failing to spend enough time researching the business idea to see if it's viable.
2. Miscalculating market size, timing, ease of entry and potential market share.
3. Underestimating financial requirements and timing.
4. Over-projecting sales volume and timing.
5. Making cost projections that are too low.
6. Hiring too many people and spending too much on offices and facilities
7. Lacking a contingency plan for a shortfall in expectations.
8. Bringing in unnecessary partners.
9. Hiring for convenience rather than skill requirements.
10. Neglecting to manage the entire company as a whole.
11. Accepting that it's "not possible" too easily rather than finding a way.
12. Focusing too much on sales volume and company size rather than profit.
13. Seeking confirmation of your actions rather than seeking the truth.
14. Lacking simplicity in your vision.

15. Lacking clarity of your long-term aim and business purpose.
16. Lacking focus and identity.
17. Lacking an exit strategy.

Andrew Montalenti:[407] **(founder of Parse.ly) Common startup mistakes**

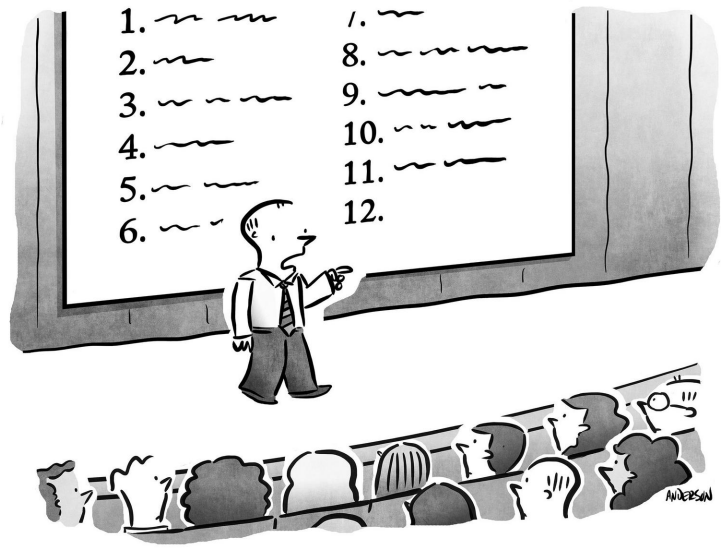
1. Marriage trouble
2. No bootstrapping plan
3. Startup as a career move
4. Refusal to change the original idea
5. Preemptive scaling
6. Growing too fast
7. Scared of code

Forbes[408]

1. Not in touch with customers
2. No differentiation in the market
3. Failure to clearly communicate value propositions
4. Leadership dysfunction, especially founder strife
5. Can't find profitable business model

Chapter 21:
Finding Fulfillment

AUTONOMY, MASTERY, PURPOSE · 生き甲斐
START WITH WHY · CIRCLES & TRAPS



"Now the twelfth and final thing you can completely change right now to be your true self..."

credit409

Many blogs, resulting in book deals, resulting in speaking careers, have been forged by taking a swing at a basic human question:

What creates a fulfilling existence?

Managers desperately ask, forced by the Great Resignation to acknowledge that people are more mobile than expected when sufficiently (un)motivated.* Enlightened leaders ask proactively, because intrinsically-motivated people do great work that is just as fulfilling for themselves as it is productive for the company—capitalism on its best day. Outwardly-impenetrable leaders ask it of themselves to conquer burn-out or to decide what to do with the balance of life that follows the societal definition of "success."

What follows is my original framework I used to answer this question for myself ten years ago when I made the decision to step aside as CEO of the (now unicorn) company I founded. I still use it myself and recommend it to others.

First, some of the best prior art on the topic.

PINK'S MOTIVATION: AUTONOMY, MASTERY, PURPOSE

Daniel Pink famously surmised** that "carrots and sticks" are poor motivators for most people. Worse, they're demonstrably counter-

* Your best talent are volunteers—they can always walk across the (now virtual) street for more money, because they're worth it, and because their resume has been incrementally improved by the experience at your company. You may protest that the grass isn't greener over there, but rather their feelings arise from the modern malaise instigated by COVID-imposed isolation from genuine relationships and the vile so-called online "relationships" fueled by toxic politics. You might even be right, but it's too late.

productive in common real-world scenarios. Instead of external motivators, blatantly designed for the benefit of the organization rather than the individual, people prefer—people *deserve*—to be driven by internal motivation, aligned with genuine personal fulfillment.

My subjective experience confirms this science-backed insight. The best people don't have to settle for anything less than personal fulfillment at work, as evidenced by the Great Resignation. And anyway it results in an organization that we are all proud to build.

Pink suggests that internal motivation arises under three conditions:

Autonomy
People are fulfilled when they decide what to do and how to do it. Counter-examples include micro-management, inflexible working conditions, and one-way command-and-control structures. Positive examples include self-managed scrum teams, work-from-anywhere-and-whenever schedules, and agreeing on the goals of the final product rather than dictating the details of what that product is and how it must be created.

Mastery
Great people want the opportunity to do great work. They want to be around other people who are doing the same. Experts enjoy deploying their expertise; novices with vim and aptitude enjoy learning and growing.

Purpose
As the janitor famously answered in 1962 when president Kennedy asked him what he did for NASA, "I'm helping put a man on the moon." Everyone—not just Gen Z—wants to be a part of something bigger than themselves. That could be a noble cause, or something more incremental but tangible, like genuinely helping another human being in their own endeavors, as one might do in a world-class customer service organization.

** in his book,[410] though you might prefer to watch his extremely popular TED talk[411]

Other work agrees with and extends these ideas.* However, I believe that to leap from Pink's original question—*What motivates people?*—to my question—*What is fulfilling?*—at least one vital component is missing: **Joy**.

生き甲斐: A REASON TO BE ALIVE

For thousands of years the Japanese have revered those who devote their lives to the mastering of a craft, having "craftsman spirit" 職人気質 (shokunin kishitsu). The samurai and the flower-arranger hold equal value, the chef and the janitor hold equal prestige, when each are whole-hearted in their endeavors. Furthermore, world-class skill is only half of the meaning of being a shokunin; you must also play a mindful and intentional role in bettering the community.

It is not only "Mastery" but also "Purpose." The Japanese have venerated 2/3rds of Pink's trifecta for millennia.

An equally ancient concept, that received a direct name in the 1960s, is 生き甲斐 (ikigai)—a motivating force that gives someone a reason for living:

> ... ikigai ... usually means the feeling of accomplishment and fulfillment that follows when people pursue their passions. Activities that generate the feeling of ikigai are not forced on an individual; they are perceived as being spontaneous and undertaken willingly, and thus are personal and depend on a person's inner self. (Wikipedia[413])

* For example, Self-Determination Theory[412] asserts that motivation arises from "autonomy, competence, and relatedness"—essentially the same thing.

This adds color to the idea of intrinsic motivation as well as that of extrinsic purpose. Furthermore, it's not an exaggeration to say that this is a reason to go on living at all:

> National Geographic[414] reporter Dan Buettner[415] suggested ikigai may be one of the reasons for the longevity of the people of Okinawa. According to Buettner, Okinawans have less desire to retire, as people continue to do their favourite job as long as they remain healthy. "Moai", the close-knit friend group, is considered an important reason for the people of Okinawa to live long. In 2016, [Héctor García and Francesc Miralles published] a book based on this concept: *Ikigai: The Japanese Secret to a Long and Happy Life*. (Wikipedia[416])

We need purpose, whether it's the Silicon Valley notion of "changing the world" or to be a great-grand-parent. To be useful, to be needed, even to be wanted, one person to another, is already a higher purpose.

One person at a time, can be all it takes, as in this serendipitous Slack exchange I had with WPGraphQL[417] founder Jason Bahl[418] (Figure 1).

A "higher purpose" might elicit an eye-roll from the skeptic, but it turns out to be useful even if you're a cold-blooded, mercenary, even authoritarian-style leader.

SINEK'S FANATICISM: START WITH WHY

Simon Sinek wanted to know why some organizations produce fanatics. Not just customers and employees, not just people who like and buy the products, but people who personally identify with the com-

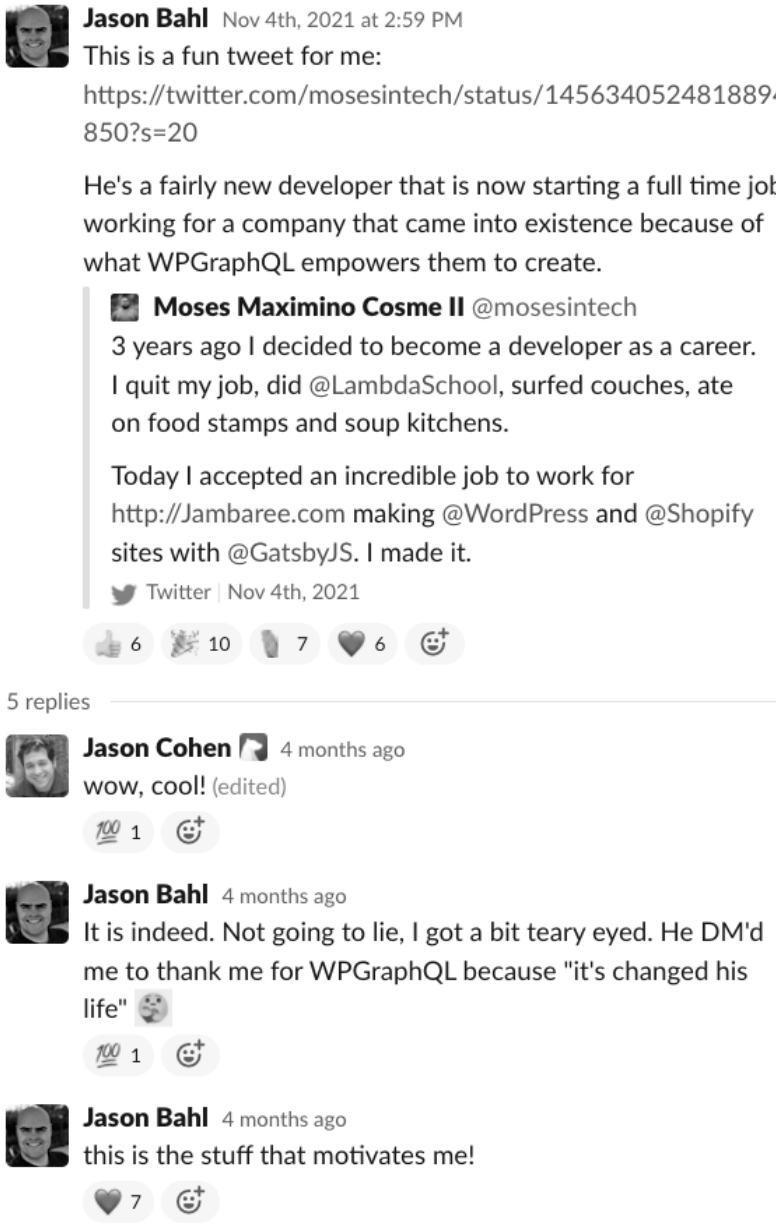
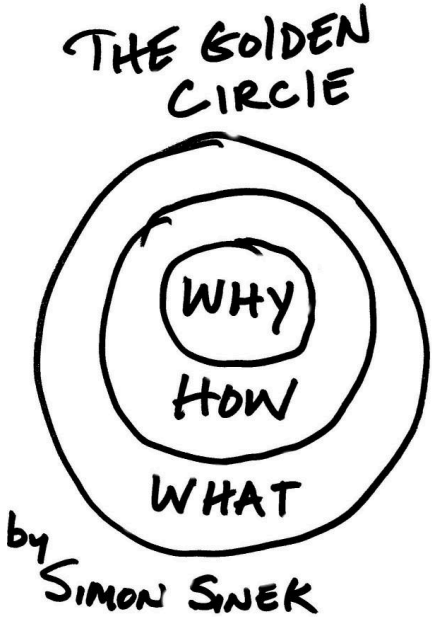


Figure 1: *(reprinted with permission)*

pany, incorporating the company's brand into their personal brand. When competitors build something objectively better, they don't switch, because for them it's not just about the features and the price. They'd work for that company if they could, and many do. As a result, employee turnover is low and productivity is high. Therefore every aspect of the business is stronger. How does this arise?

Sinek sums up the answer: These companies **Start with "Why."** Meaning: These organizations have clear, simple, compelling *raison d'être*, a reason for being, something they stand for, something they would never contravene with their actions, even if it hurts sales or profitability.

It could be a "higher purpose," like Patagonia's incontrovertible mission to save the Earth,* or SpaceX's mission to make humanity a



credit[419]

* Beyond the obvious—sustainable practices, an outdoor-worshiping culture—they have a formal company policy[420] to bail employees out of jail if arrested while protesting peacefully.

multi-planetary species, or the Gates Foundation project to save millions of lives by eradicating malaria.

But it doesn't have to be grandiose or holy to be motivating. Sinek frequently returns to the example of Apple's "Think Different" campaign; this taps the personal brand of those who are (or want to be seen as) independently-minded, creative, creators, iconoclasts, and artists, including artists embedded in fields like software development and engineering. Or Salesforce with its "1-1-1"[421] or WP Engine with its "Engine for Good"[422]—thoughtful, systematic ways of giving back to the communities they are a part of (without pretending to single-handedly "change the world"). Or Rackspace with their "fanatical support," transforming the traditional status of Customer Service from a begrudged cost-center hell-bent on cost-reduction, elevating it instead into the honor of serving others, and a critical product differentiator in a market that is otherwise commoditized.

> *"When people are financially invested, they want a return. When people are emotionally invested, they want to contribute."*
>
> *—Simon Sinek*

All of these companies were created by and are still run by Gen-X-and-older. So, wanting a higher purpose isn't just "a Gen Z thing," though clearly younger generations do, on average, talk about this more, and more conscientiously. In any case, the freshest talent and the future leaders of the world are Gen Z, so even a leader unsympathetic to "causes" should realize that having a higher purpose is more useful than not having one.

This confirms and clarifies the "Purpose" component of the Pink Trifecta. Sinek approached it from an analysis of fanaticism and loyalty to organizations, and arrived at the same place.

COHEN'S CIRCLES & TRAPS: JOY, SKILL, NEED

It is possible to be empowered to work how you want (Autonomy), to be leveraging your skills and expertise (Mastery), and to be proud of your role in a cause (Purpose / Why), and yet *still dislike every day of your existence*. More than contentment (ikigai), you need *Joy*.

Not only is this possible, it is common. There's the classic example of the startup founder who wakes up six years into the journey, realizing she's been surreptitiously brought to a boil,[423] burned out, dreading each day, drinking too much "to turn my brain off so I can sleep" but actually because she's deeply unhappy:

> Each morning for the past couple months, my first thought has been "What could today be like if I didn't work here?" I drift off into exploring what it would be like to work at WalMart, or the construction site outside, or as a grocery store bagger. It seems so stress free. This morning, I locked myself in the bathroom with the shower running (don't want wife to know) and cried my eyes out. I haven't cried in many years. It felt great, but only for an hour.
> —Anonymous founder[424]

Eight years ago I created my own framework for understanding how I could avoid this burn-out trap at WP Engine,* especially when making decisions that are contrary to ego. The ego says "Being the CEO is the best." Silicon Valley says "The founder being the CEO is

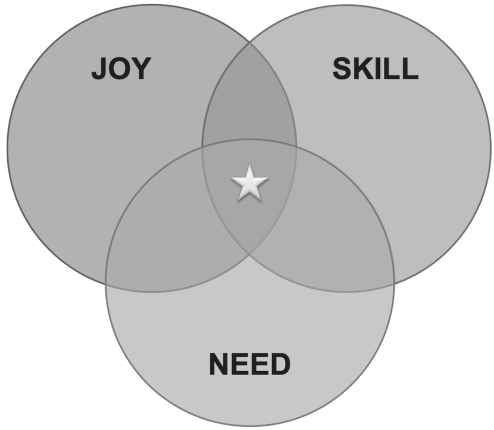


Figure 2: Someday I'll make a Venn diagram where "the middle" isn't the answer, but today is not that day.

the Only Way."** So it's hard to convince the ego to let go of being the CEO, as I did at WP Engine, even if being the CEO creates unhappiness.

Here's the insight: Not only do you need *all three* of the following components, but any two alone create a specific trap (Figure 2).

Joy
You love doing it. When you do it all day, you forget to eat and pee. At the end of a long day of doing it, you still want to do it.

Skill
You're great at it. Your work is so good, even *you* are proud of it. It stands out, and others notice. Those who don't understand how much effort you expended say: "You're a natural."

---

* Whereas I did burn out at my previous company Smart Bear. Even selling the business didn't immediately fix the problem (p. 1005), although it was the beginning of the answer.

** Typically citing Apple, Facebook, and Amazon while ignoring counter-examples like Google, LinkedIn, and Intel, although there do exist thoughtful arguments that embrace those exceptions.[425]

Need
The company needs it done. It's a top-three priority. Doing it well means a critical part of your strategy will succeed. Not doing it is crippling.

Having any two without the third creates a well-defined yet common trap. It's instructive to understand the traps, because it can *feel good* to be in the trap:

Trap: Joy + Skill – Need = Useless Flow
At the intersection of Joy and Skill is "being in the zone," a.k.a. Flow.* Wonderful! Unless you're working on something the company doesn't need done. Being in flow is intoxicating, and does "recharge the batteries," but it's unproductive. Continuing the example of the burned-out founder living in this trap, there's no one willing or able to tell you to your face that you're not contributing to what the company needs, so you stay in your happy place. A classic example is the technical founder writing code instead of making sales, fixing the website, handling the accounting, or hiring the next great team member.

Trap: Joy + Need – Skill = Indulgent Failure
The company needs to begin advertising. You've never done AdWords before, but you always wanted to try it, and anyway it'd be fun to learn something new. You might hire someone to do it later, but not now, because how do you hire and manage someone else without understanding the job yourself? So you make the AdWords campaigns. And since you're unskilled at both marketing and AdWords, you waste three months. You might even erroneously conclude "AdWords doesn't work" because that's an easier conclusion than "I don't know what I'm doing." If the company needed advertising so badly, you did a disservice to the company by indulging your desire to "play with it," because now months have passed without accomplishing what needed to be done. You tell yourself that *now* you can hire that person, because three months of flailing somehow makes you an "expert." As founder, no one will contra-

* Flow[426] is when a person is fully immersed in an activity for an extended period of time. Time passes unwittingly, performance is at maximum, and it is universally described as pleasurable.

dict you, so you indulge, and the company falls that much further behind.

Trap: Skill + Need – Joy = Burn-out
This is classic burn-out. When you do the work all day, you feel drained and exhausted rather than energized (as you would if it were Flow = Skill + Joy). You do the work, because the company needs it done. You do the work, because you are undeniably great at it. Even though you hate doing it, you'd rather take it on yourself rather than foist it on others, whether because you want to "protect them from the drudgery,"* or because you believe they can't do as good a job as you can, or because you can't afford to hire someone. Because you create great results that the company needs, it doesn't look like a problem—not to you, nor your team. But because you dislike it, you grow to resent it, and eventually you can't face it, and you're finished. Many startup founders agree.[427]

I created and used this framework nearly a decade ago, to recognize the wisdom of changing roles from the CEO of the company I founded (WP Engine[428]) to the CTO, so that both our new CEO and I could operate in the center of our Venn diagrams. The thousands of people who have since worked at our company concur that this was a fantastic decision, both personally for the two of us, and for the success of the business, which is now a unicorn and an iconic landmark in Austin, Texas (Figure 3).

This framework is not a replacement of Pink's model; it is compatible. It adds the missing "Joy" component, while reinforcing "Mastery" with the label of "Skill." It lacks "Autonomy," however, perhaps because I created it with the founder in mind—a person who definitionally possesses autonomy, even to their detriment. "Need" is more tactical than "Purpose," really about being *useful*.

Therefore, my recommendation is to identify that higher purpose, as described by "Start with Why" or ikigai, and fulfill your own part in that purpose at the center of the three circles.

---

* No one wants the boss doing all the drudgery, especially if the boss visibly hates doing it. Some people actually *like* the thing you hate, and you're preventing them —and you—from being happier.

Figure 3: Co-winning E&Y's Central Texas Entrepreneur of the Year Award in 2017 with our inimitable CEO Heather Brunner,[429] who I frequently refer to as "the founder who joined three years in." Copying how Reid Hoffman describes Jeff Weiner at LinkedIn, it is neither an exaggeration nor an unearned appellation.

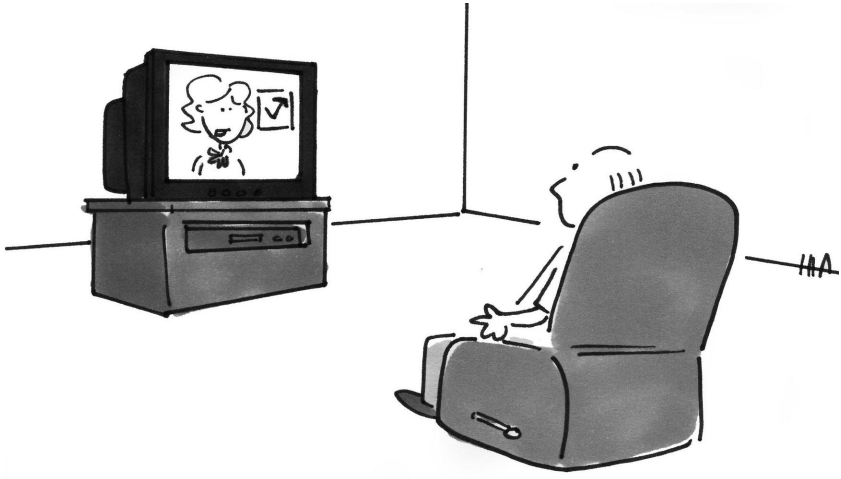
When we intentionally create a work environment where others have a chance at fulfillment, we're already improving the world substantially. When we care enough about others to find out what fulfills them, we can help that materialize. When we're closer to our own center of fulfillment, we can better serve others.

How can you find out who you are, what drives you, what stops you? Here's the system I've used for myself and others (p. 569), as well as how I figure out "who we are" as a whole product or company.

I wish you luck on your journey!

Thank you to Jason Evanish[430] *for providing feedback on earlier drafts.*

Chapter 22:
In command



"Stocks rose slightly in early trading, then plummeted on news that stocks rose slightly in early trading."

credit[431]

When a founder or CEO or team leader is "in command" of her business, I don't worry about that business, regardless of its challenges.

Being "in command" doesn't mean you always meet your own expectations for yourself or your team. It means you *have* expectations of yourself and your team, and when you don't meet them, it bothers you, and you react.

Being "in command" doesn't mean the metrics are always healthy, nor even that they're moving in the right direction. It means you've already thought about what to measure (p. 645) and you're watching the things that really matter. When metrics go awry, you're the first one to identify that, and are already taking action in proportion to their importance and how far off the mark they are.

Being "in command" doesn't mean the product has no bugs. It means you have a list of 100-1000 bugs, and the reason you're not working on them is that you're working on the top three that matter most.

Being "in command" doesn't mean you're always the smartest person in the room. It means you've hired a team of great people who can challenge you and bring diverse perspectives to the table.

Being "in command" doesn't mean you never make mistakes. It means you're proactively on the look-out for mistakes, and call them out when you find them, so that now you can rally many people in finding solutions, and make a change.

Being "in command" doesn't mean you never feel overwhelmed. It means you recognize when you're overwhelmed, take steps to manage your stress, and seek support when needed.

Being "in command" doesn't mean there are no problems with any team member. It means you've already talked to that person about it, and you're working with them to change. Whether that means changing and staying, or changing by leaving. It means you're proactively and intentionally solving for fulfillment (p. 399) for the whole team, which includes yourself.

Being "in command" doesn't mean there are no existential threats looming over the business. It means you've articulated what those are

because you sought them out and faced them head-on (p. 657), and you're mitigating the more likely (p. 997) ones with a few, logical actions.

Being "in command" doesn't mean customers aren't leaving due to lack of features. It means you've identified what those features are, which ones are consistent with your strategy, which ones lead to the ultimate benefits the customer is seeking (p. 259), and that you're working on one or two of them right now.

Being "in command" doesn't mean you're always popular. It means you're willing to make tough decisions, even if they're not always well-received, because they're in the best interest of the customer, the team, and the business.

Being "in command" doesn't mean you're constantly telling people what to do. It means you're acting as an editor, not a writer, unless being a writer is necessary to make progress. And if *that* is happening constantly, that you make a change on the team so that it's *not* happening constantly. That might mean changing someone else, or changing yourself.*

Being "in command" doesn't mean your strategy is completely correct. It means you are adhering to the characteristics of great strategy (p. 489). You have one, that is written down, that a normal person can understand, that people genuinely try to follow. When strategic flaws become apparent, you write those down too, and work on one or two at a time, updating your strategy calmly but purposefully.

Being "in command" doesn't mean surprises never occur. It means when they do occur, it's not due to negligence. It means something actually changed, or an important new fact was uncovered. Even better, if you were intentionally seeking new information (p. 239), and thus seeking surprise.

---

* If you're constantly having to do someone else's work, that's your fault. Either you've hired incorrectly, which is your fault, or you have hired correctly and you're not allowing them to do their job, which is your fault.

Being "in command" doesn't mean you win every sale. It means you seek to understand the patterns in the wins and losses, and when those intersect with something consistent with your strengths and strategy and thus where you *should* win, you make changes so you can win the next such deal.

Being "in command" doesn't mean every A/B test is a success. It means you're running A/B tests and honestly assessing the results, so that you improve over time.

Being "in command" doesn't mean you don't have a long backlog, most of which you'll never get to. It means you've identified the few Rocks (p. 221) that are your bets on how to win strategically, and you have a system for prioritizing the rest (p. 603), which also means doing almost none of the rest.

Being "in command" doesn't mean you know everything about your target customer. It means you are actively seeking out (p. 239) what they say and do, so you can understand them better, so you can make better decisions.

> "Well when events change, I change my mind. What do you do?"
> — Paul Samuelson, winner of 1970 Nobel Prize in Economics, about how his models of inflation during WWII kept changing over time, and he was criticized for "not being able to make up his mind."

Being "in command" doesn't mean you never change your mind. Indeed, a mind that never changes, is very likely wrong. It means you wrote down what you think, so that it's more obvious when your mind needs to change, and then you communicate that change—both what and why. Even if just to yourself.

Being "in command" doesn't mean you do everything that stakeholders or even executives ask you to do. It means you listened with curiosity and empathy to what has been asked, having a system

(p. 711) to reject bad requests and prioritize good requests, communicating that result and the rationale behind it.

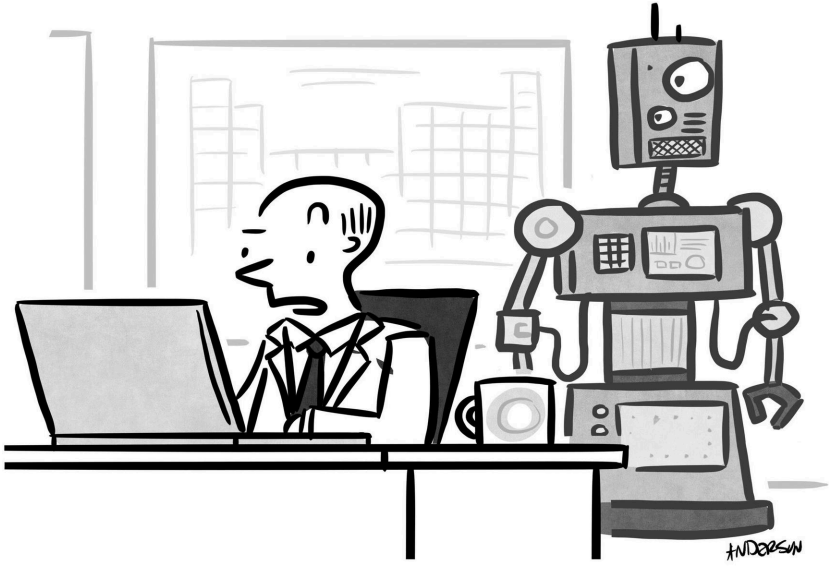
Being "in command" doesn't mean you're "in control" of everyone and everything. It means you're proactively seeking the truth, working rationally, and communicating.

This is how you create autonomy, coupled with accountability, with realistic expectations.

Strive to be "in command."

Chapter 23:
AI startups require new strategies: This time it's actually different

DISRUPTION THEORY · DIFFERENTIATION · DATA
TALENT · MARKET



"Hey, what's this 'uprising' on the calendar for Friday?"

credit[432]

Startups must leverage their specific advantages they have over incumbents (p. 295) to overcome the massive barriers erected by those incumbents: Mature products, mature distribution channels, brand, trust, optimized organizations, and cash in the bank.

However, in the AI revolution many of those advantages are unavailable. Some even become a disadvantage. This is substantively different from the other two major technological revolutions of the past twenty-five years.*

Specifically:

Disruption Theory & risk-aversion don't apply
Incumbents typically cede market space to startups wherever there's new, unproven technology or a new, unproven market, especially in spaces where they can't use past data to predict the future. But in AI, they're rushing to embrace new technology and uncertain markets, spending historic amounts of money and time.

Incumbents aren't failing to innovate
Typically the race is "whether the startup gets distribution before the incumbent gets innovation (Alex Rampell[433])." But in AI, the incumbents already have the innovation (whether through closed APIs or open source), while startups struggle as mightily as ever to find distribution. Perhaps an even greater struggle, as every market is over-saturated with new startup competitors, some with massive funding.

Incumbents have the data
"There's no AI strategy without a data strategy," the (now often-repeated) saying goes. For training, testing, benchmarking, and features, you need data. Incumbents have it or can afford it; startups are at another disadvantage.

Great talent are happy at incumbents
While there will always be fantastic people who only want to be at a small company, there are even more people, including the top AI and

* The Internet and mobile devices. Not blockchain, at least not yet—that has had almost no effect on businesses despite the application of billions of dollars over more than a decade.

software engineering talent in the world, who are getting paid above-market rates to work on the most exciting projects, with healthy budgets, with data to leverage, and with the ability to impact huge numbers of customers quickly without having to do marketing, sales, support, or accounting.

The so-called "AI market" is not what you think it is
People say "the AI market will be trillions of dollars so there's enough for everyone," but there's no such thing as "the AI market" unless you're competing directly with OpenAI. The market for chatbots and SEO tools is the same market as before, now with stiffer competition.

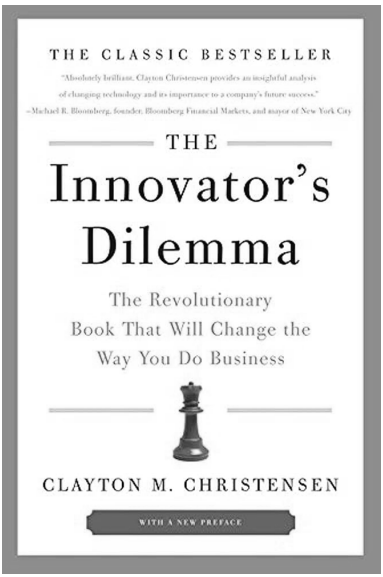
The typical startup strategies are wrong because typical strategies rely on the typical advantages of David against Goliath, which are largely untrue in AI. If strategy is "how we will win," (p. 489) then the "how" must be different with AI.

Few AI startups have embraced this reality, and most will fail as a result. It is easy to create an AI startup, gain attention, and generate excitement, but it is much more challenging to compete and win.

The remainder of the article details and defends this case.

DISRUPTION THEORY DOES NOT APPLY TO AI

The Innovator's Dilemma is not a dilemma this time around.



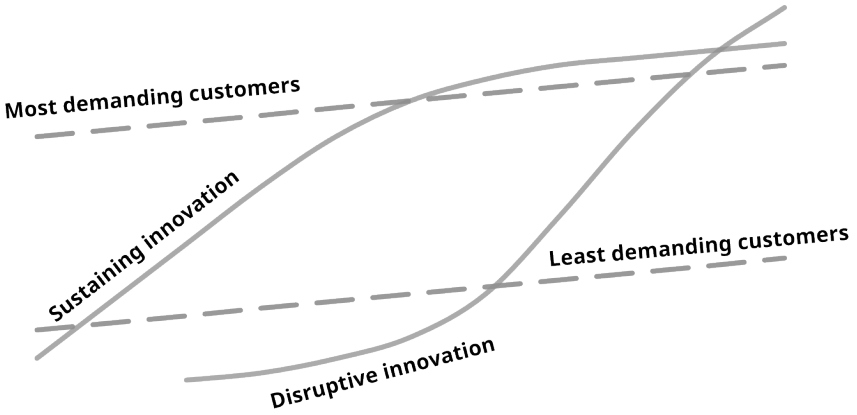
Disruption Theory explains how little upstarts leverage (p. 543) new technology to topple incumbents, even though the incumbents see it coming and act in their rational self-interest.

The theory goes: New technology is often (a) better in some ways, (b) worse in some ways, and (c) cheaper than existing technology. The combination of (a) and (c) can overcome (b). The classic example given in the book is the original transistor radio, whose the sound quality was far worse than the incumbent, beautiful-to-look-at, rich-sounding, furniture-sized-and-weight RCA radios. However, the advantage of the transistor radio is that it is portable, a feature made possible because transistors use so little energy. A radio at the beach that's staticky and a tinny is a lot better than not having a radio at all. So, upstarts were able to sell radios that were worse in a very important way (p. 891), but also cheaper and with the advantage of mobility.

Of course, over time, the transistor radio improved its sound quality, and this is the other factor in Disruption Theory: The technology that starts out worse becomes better over time, and yet is still cheaper and retains its positive qualities. Then the new technology fully disrupts the incumbents, who had been looking on, thinking "we're the ones who have great sound, so we're safe. We're not even really competing with those toy radios."

In AI, the conditions of Disruption Theory are not present.

The incumbents do not look at AI and say "this technology is cheaper but worse, so we will ignore it." They're doing just the opposite. They say "this technology is groundbreaking and if we don't embrace



credit 434

Disruptions start "worse," but then become even better than incumbents

it, we will become irrelevant" and "our shareholders are demanding that we invest huge percentages of our annual revenue to do it."

Therefore, startups who normally depend on incumbents ignoring new technology and new markets, cannot depend on that in AI, and therefore are competing directly with well-funded incumbents who are selling to their existing customer bases.

DIFFERENTIATION WHEN EVERYONE HAS THE SAME TECHNOLOGY

The "hard tech" in AI are the LLMs available for rent from OpenAI, Anthropic, Cohere, and others, or available as open source with Llama, Bloom, Mistral and others.

The hard-tech is a level playing field; startups do not have an advantage over incumbents.

There can be differentiation in prompt engineering, problem break-down, use of vector databases, and more. However, this isn't something where startups have an edge, such as being willing to take more risks or be more creative. At best, it is neutral; certainly not an advantage.

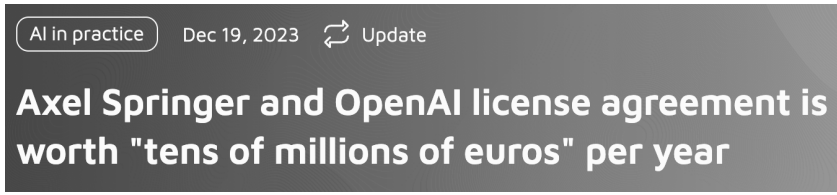
In a market where everyone has access to the same core technologies, simply matching the capabilities of established players is not a winning strategy. This doesn't mean it's impossible for a startup to succeed; surely many will. It means that you need a strategy that creates differentiation and distribution, even more quickly and dramatically than is normally required.

THERE IS NO AI STRATEGY WITHOUT A DATA STRATEGY

Training AI models requires data. Whether you're training existing models, developing models from scratch, or simply testing theories, high-quality data is crucial.

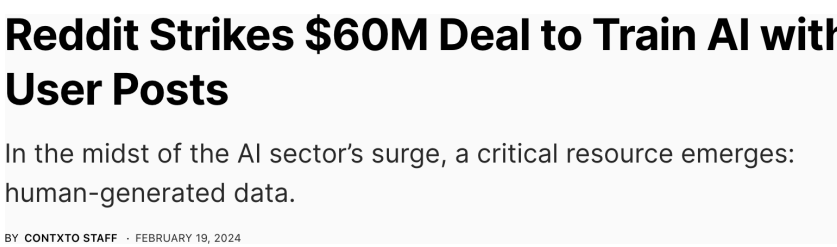
Incumbents have the data because they have the customers. They can immediately leverage customers' data to train models and tune algorithms, so long as they maintain secrecy and privacy.

For instance, Intercom's AI strategy is built on the foundation of hundreds of millions of customer interactions. This gives them an advantage over a newcomer developing a chatbot from scratch. Similarly, Google has an advantage in AI video because they own the entire YouTube library. GitHub has an advantage with Copilot because they



credit[435]

Figure 1



credit[436]

trained their AI on their vast code repository (including changes, with human-written explanations of the changes).

Incumbents can also afford to pay for data (Figure 1).

AI will continue to become cheaper, better, and faster in the next few years. Getting access to huge amounts of training data will not; in fact, just the opposite as the companies who possess that data have already learned how valuable it is.

GREAT TALENT ARE HAPPY WORKING AT INCUMBENTS

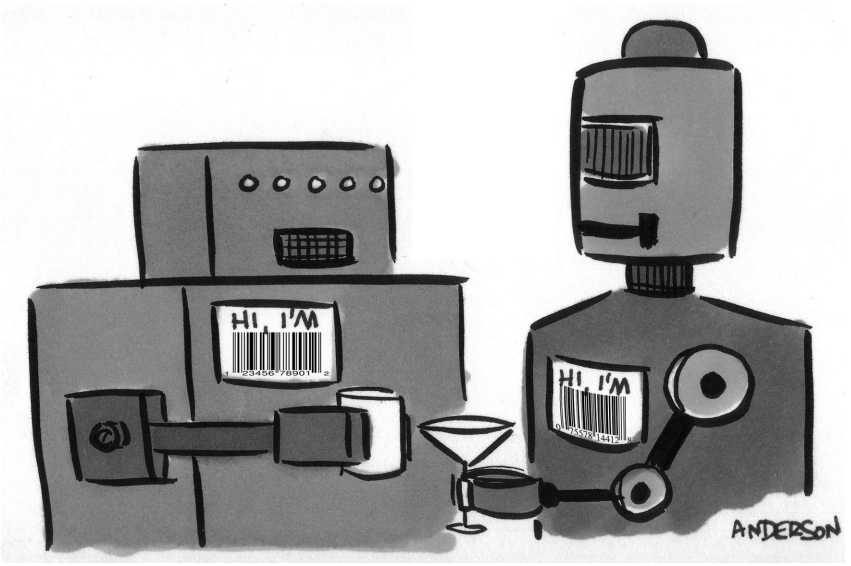
Traditionally, startups have attracted dynamic and innovative talent with promises of autonomy and wealth potential, outshining rigid incumbents.

However, the landscape shifts when it comes to AI. While there will always be individuals preferring the startup environment, the allure of working on AI at an incumbent is equally strong for many, especially pure computer and data scientists who, more than anything else, want to work on interesting AI projects. They get to work in the code, with a large budget, with all the data, with above-market compensation, and a built-in large customer base that will enjoy the fruits of their labor, all without having to do sales, marketing, tech support, accounting, raising money, or anything else that isn't the pure joy of writing interesting code. This is heaven for many.

Many people want to create or join a startup of course. But startups don't have an inherent advantage for attracting talent; at best this is neutral again.

THE "MARKET" IS NOT WHAT YOU THINK: AI IS NOT A LARGE, GROWING MARKET

"AI will be a multi-trillion-dollar market" people say. AI is surely the definition of a large, growing market, with many orders of magnitude of growth in its future. Which means it's perfect for startups.



credit[437]

Except these are nonsense statements. A "market" is a set of buyers, with sufficiently similar needs, constraints, and goals, that the same product can be sold to all of them. The "AI" market consist of companies like OpenAI and Cohere, and indeed that is a large and growing market, but unless you're competing directly with them, you're not "in the AI market."

A chatbot is in the chatbot market, and an SEO tool is in the SEO market. Adding AI to those tools is obviously a good idea; indeed companies who fail to add AI will likely become irrelevant in the long run. Thus we see that "AI" is a new tool for developing within existing markets, not itself a new market (except for actual hard-tech AI companies).

AI is in the solution-space, not the problem-space, as we say in product management. The customer problem you're solving is still the same as ever. The problem a chatbot is solving is the same as ever: Talk to customers 24/7 in any language. AI enables completely new solutions that none of us were imagining a few years ago; that's what's

so exciting and truly transformative. However, the customer problems remain the same, even though the solutions are different.

That said, AI can reignite growth in existing markets. Companies will pay more for chatbots where the AI is excellent, more support contacts are deferred from reaching a human, more languages are supported, and more kinds of questions can be answered, so existing chatbot customers might pay more, which grows the market. Furthermore, some companies who previously (rightly) saw chatbots as a terrible customer experience, will change their mind with sufficiently good AI, and will enter the chatbot market, which again grows that market.

Still, the right way to analyze this is not to say "the AI market is big and growing" but rather: "Here is how AI will transform this existing market." And then: "Here's how we fit into that growth."

Even so, it sounds a lot easier for an incumbent with a trusted brand and existing customers and all the data, than for a startup. Not impossible for a startup, again the startup is at a special disadvantage that it doesn't normally have, especially when the startup is entering a new niche, a new market, or is in a market which is organically large and growing.

AI is different from previous technological disruptions. The dynamics between startups and incumbents are fundamentally different, and **that means you need a different strategy**.

If an AI startup has a strategy that looks like a strategy a non-AI startup had five years ago, that may very well be the wrong strategy.

This doesn't mean you shouldn't make an AI-based startup. In fact, all new startups probably do need to include AI.

It means the usual startup strategies might not work, so you need to think deeply and have a different point of view about how you're going to win.

Chapter 24:
It's a torturous chaos until it isn't

When you hear "explosive startup growth," a few darlings come to mind. Facebook, Slack, now OpenAI (Figure 1).

I wonder... what did they all look like prior to reaching $1M ARR, i.e. prior to "year 1" on that chart?

HubSpot is a good example, because all their data is public. They are a proper juggernaut of growth; as of the beginning of 2024, at a whopping $2B ARR, they're still growing quickly (Figure 2).

Shifting eight years earlier, the curve looks similar, just with smaller absolute numbers (Figure 3).

Shifting earlier still, it still looks similar, but now we've arrived at the crux of our investigation: The beginning (Figure 4).

Wait, how long did it take them to get their first 500 customers? Two years?

Yes. In fact, after ending 2006 with just 3 (three!) customers, they ended the next year with only 48 (source[442]). That's less than one per week. For a year.

That does not sound like a juggernaut of growth. In fact, that sounds downright bad. Like, maybe you should shut the company

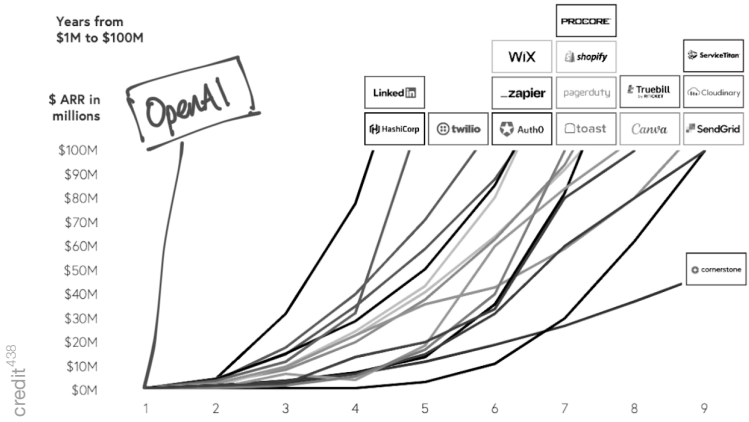


Figure 1: My own startup WP Engine was in the middle of this pack; a good place to be!

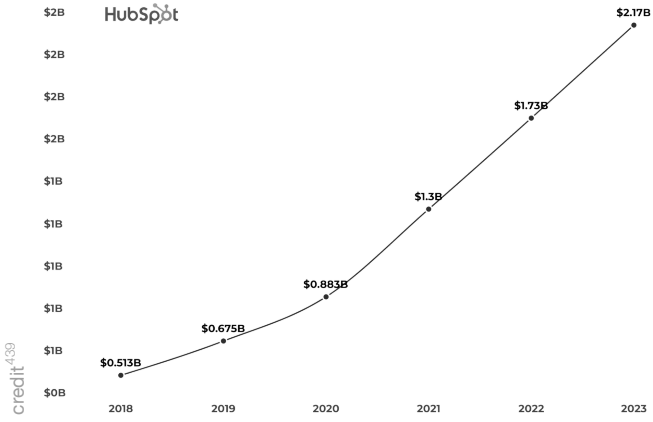


Figure 2

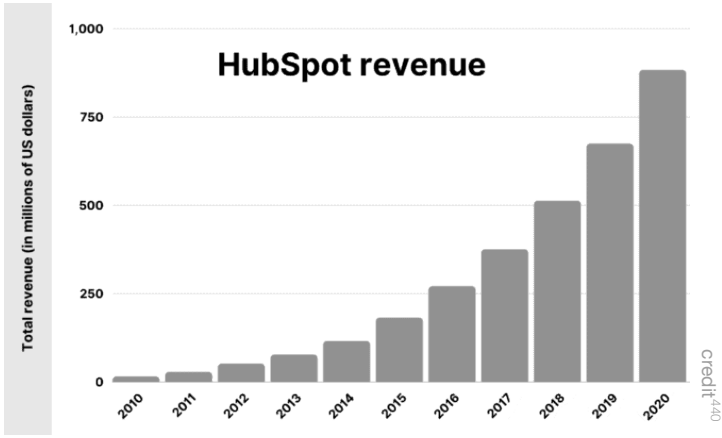


Figure 3

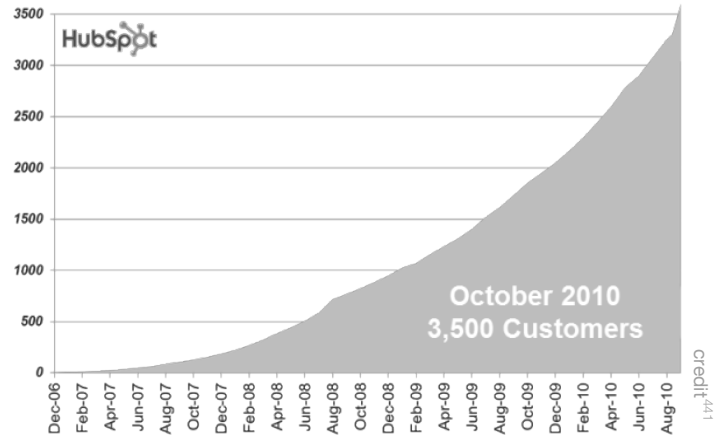


Figure 4

down because it's "obviously not working." Or at least, there should be an existential struggle over whether it is wise to continue (p. 159).

What was going on during those first years? What it was *not*: an obviously bright future with a clear vision validated by a torrent of customers with open wallets.

So, what *was* it?

A barely-controlled chaos, awake at 2:43am worrying how to make payroll or whether the numbers will look good enough by next spring to raise another round? Unsure which product features to double-down on and which to kill, worried the wrong choice would tank the entire company? Staying bright and cheery on the outside for the press, customers, investors, and employees while that one horse-hair tenuously suspending Damocles' Sword is looking awfully thin and worn?

Of course it was. It's always like that.

Then again, the story is the same when the company "runs off a cliff without leaving skid-marks," never reaching Product/Market Fit (p. 335); possibly never having extracted a single dime from a single person. The one that really *should* have shut down.

The same struggle, same uncertainty, same near-impossibility, same mess.

So how do you tell the difference between the tortuous chaos that leads to unimaginable success and that which leads nowhere at all?

I'm not sure you can.

I mean, there are some questions you can ask yourself (p. 159). But:

Objectively, what's the difference between Basecamp v1.0 and any number of knock-offs? All of them "the tool we wish we had, so we built it ourselves (p. 533)," all of them simple, all of them usable by relatively non-technical people, all of them web-based, all of them missing half the features that any one customer wished it had.

Objectively, what's the difference between StackOverflow and the twenty copycat Q&A websites and thousands of forums? Why was Quora still able to stand out? Could you measure this difference? Could you plot it on a chart?

One thing you know for sure—what *doesn't* explain it is a "feature comparison chart." Or a SWOT.[443] Or financial metrics. Or social

media. Or whether they raised money. Or whether they went to Stanford. Or whether the market was already there. Or whether the competition was already there.

Or anything else. Even with my roadmap for Product/Market Fit (p. 9) in hand, there are exceptions to everything. I say in that article that there needs to be founder-fit, but the founders of AirBnB weren't experienced in hotels or hospitality and the founders of Uber weren't experienced in car services. And that's just one assertion among dozens from that one article; there are exceptions to every "rule."

Take Peldi,[444] founder of Balsamiq Mockups.[445] He blogged a bit, but unlike StackOverflow he didn't have a head-start with an online presence, and English is his second language. He lives in Italy, not surrounded by an American startup culture. Prior to entrepreneurship he was a lifer at Adobe, so that didn't prepare him for bootstrapping a company from scratch.

His growth chart in year #1 (Figure 5).

How do you think Peldi felt during those first six months? Do you think the anguish subsided quickly? Or ever? Do you think Peldi was still sure his ideas and execution were good in June? Or even after-

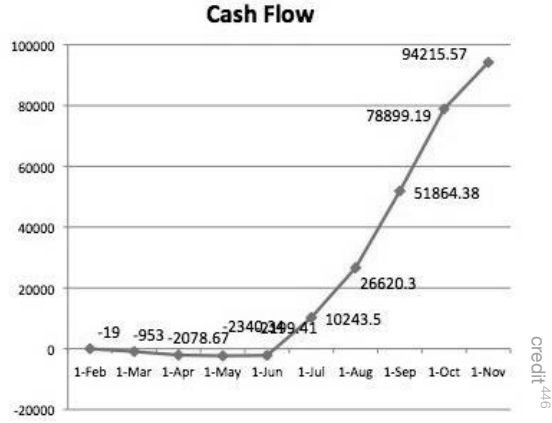


Figure 5: Balsamiq Markups revenue, 2008

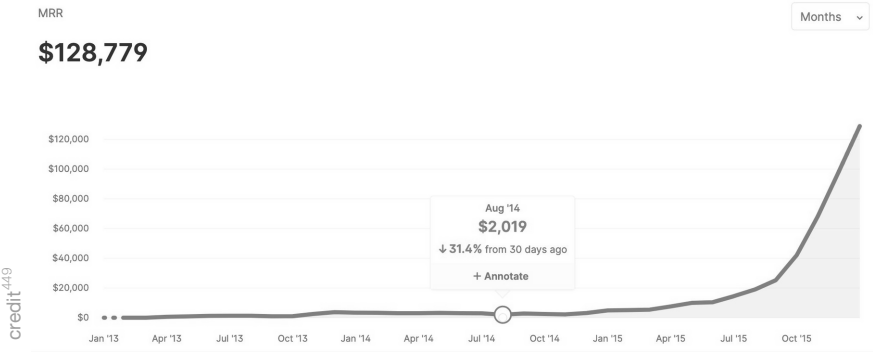


Figure 6: ConvertKit's early MRR growth

ward—this was one-time revenue, not recurring, so it could stop at any time.

Or more recently, Nathan Barry[447] with Convertkit.[448] How did it feel in January 2015, more than two years in and almost nothing to show for it? (Figure 6)

Or more recently still, Tim Bennetto[450] with Pallyy,[451] with two years of making less than $3k/mo in MRR, and then it changed (Figure 7).

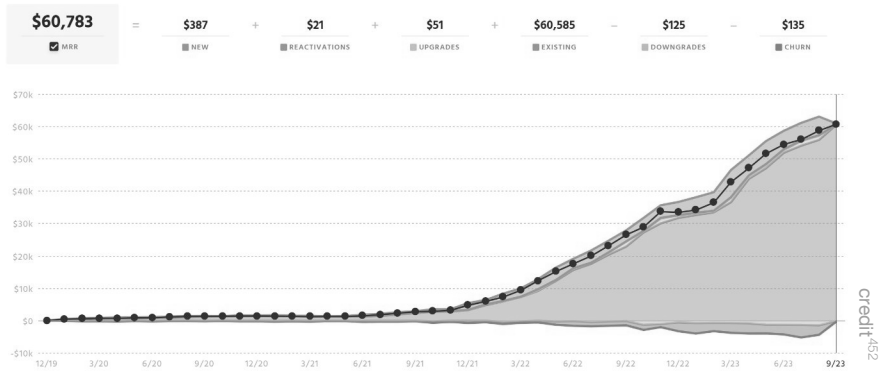
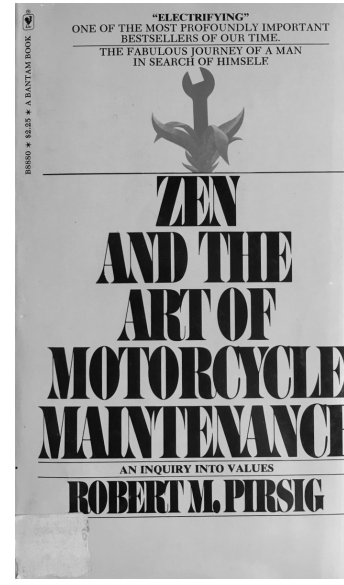


Figure 7: Pallyy's four-year MRR

And it's not just startups.

Robert Pirsig's book *Zen and the Art of Motorcycle Maintenance* was rejected by 121 publishers. How do you keep going after "just" 100 rejections? After 20, even?



Then it was published by William Morrow & Company in 1974 and stayed on best-seller lists for decades, with total sales of over 5 million copies. For comparison, Dr. Seuss's *Green Eggs and Ham* has sold 8 million copies.

Speaking of Dr. Seuss, he was rejected by 27 publishers before *Green Eggs* was published. Stephen King's *Carrie* was rejected 30 times before Doubleday took a chance on it. J.K. Rowling was rejected 12 times. "I was the biggest failure I knew" she said of herself during that time, before she went on to sell 450 million books and be worth over $1B (and became famous for things other than the world of wizards).

> *"I've missed more than 9000 shots in my career. I've lost almost 300 games. 26 times, I've been trusted to take the game-winning shot and missed. I've failed over and over and over again in my life. And that is why I succeed."*
>
> *–Michael Jordan*

Of course, your struggles might indeed be in vain. The vast majority of authors never succeed. Nor startups. Nor new product lines (p. 793) inside established companies. Most of the time, it doesn't work; that's the brutal truth.

But this I do know:

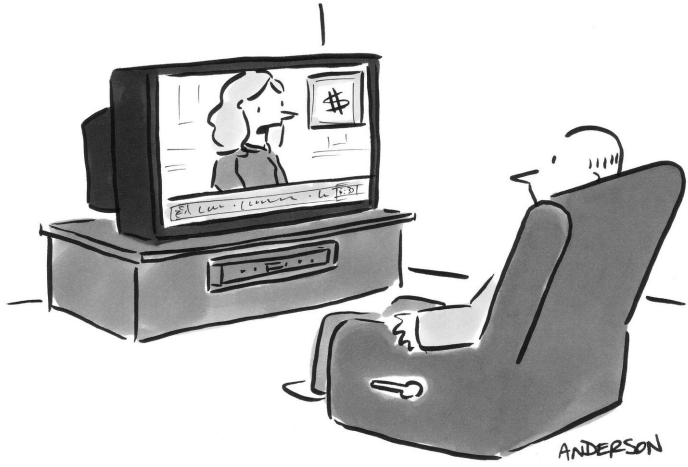
The fact that you're in over your head, that you almost cannot will yourself to continue (p. 737), that you're completely in the dark, that you're working yourself to an early grave (p. 1547), that you seem to slide two steps back for every one forward (p. 159), that nothing's ever good enough (p. 1433), that your friends and family can't understand why you're turning yourself inside out with no apparent progress, that you're supposed to be enjoying the journey (p. 861) but you're not, that you yourself doubt whether you're even capable of this (p. 457)...

These things mean you haven't yet succeeded, but neither do they prove that you're failing.

It's always like this, until it isn't.

Chapter 25:
Sometimes never compete on price

"NEVER COMPETE ON PRICE"
LOW-PRICE STRATEGY · BUT FINANCING?



"Due to recent economic conditions, picture worth has dropped to an all time low of 842 words."

"Never compete on price."

So we are told. But customers want low prices, and it worked for Costco, Southwest Airlines, Vanguard, IKEA, Amazon, Walmart, McDonald's, H&M, Dollar General, TJ Maxx, and many others. So why are we told not to use a low-prices to win?

When two products are fundamentally different—in features, integrations, complexity—customers often have no real choice between them. A large enterprise must use software with single-sign-on, security controls, compliance features, and the ability to scale to tens of thousands of users with roles and permissions and integrations with corporate systems.

For a small business, those features make the software harder to use. And no one wants to pay for things they don't need. So, Enterprise- and SMB-targeted products differentiate on capabilities and user-experience, not on price.

But consider two products that solve the same problem for the same customer segment in largely the same way, with largely the same features. The sales team highlights minor differences that matter to maybe 10% of users. Each founder is certain that theirs is the Only True Way, but everyone else has a hard time distinguishing them. Price, then, is the last remaining thing they can compete on. They are two gas stations across the street from one another; the main difference is the number on the big sign.

So, one company drops its price by 20%, and starts winning market share. The other, having no other way to compete, does the same. This continues through "special offers" and permanent cuts and "branding" and kick-backs to influencers until there's no profit left. Both com-

panies end up poor, and in case your response is "Good! Companies are evil and don't deserve profits!", remember this also means they're unable to invest in product development or customer service; thus the products and companies stagnate and then deteriorate, which in turn ruins the customers' experience. Low price comes at a price, to consumer and company alike.

The math is even worse. A 20% price cut needs a 25% increase in number-of-customers to maintain the same amount revenue.* It's worse again with profit; even a seemingly-minor 10% price reduction can easily mean a 50% profit reduction**—terrible!

This math is why pundits (including me) constantly admonish bootstrapped founders to "raise your prices." Low prices leave room for neither reinvestment nor profit. New founders price their products low for the wrong reason: They don't know how to set prices, and they know their initial product isn't very good, so instead of making it simple and lovable (p. 101) and winning a reasonable price from their ideal customers (p. 317) who would be happy to pay it (p. 275), they sheepishly charge a low price, attracting the worst kinds of customers (the ones who can't afford a proper solution, but occupy the greatest amount of time in tech support), and dramatically increasing the number of customers they must accumulate before they can quit their day job, all with no extra money for marketing or sales or design or product development. This is the wrong decision, made in fear and ignorance.

So, is it true? "Never compete on price?"

---

* Charging 80% of the price at 120% purchase volume multiplies to 96% of the revenue you had. Percentage losses always require even-larger-percentage gains just to get back to where you were. This asymmetric effect also appears in the math of becoming more productive (p. 925).

** Suppose the company charges $100 for a product and makes $20 profit after all costs. Reducing the price to $90 (a 10% price reduction) reduces profit to $10 (a 50% reduction from $20). This is exactly why established companies cannot compete on price (p. 1061) on their main product, and thus is an Achilles' heel for startups to exploit (p. 295).

Well, no. Of *course* it can be an excellent strategy to have the lowest prices.

There are many examples of wonderful companies, with amazing products and happy customers, and even happy shareholders, with profits, growth, and longevity, where "low prices" is a critical strategic component.

Jeff Bezos famously quipped: "Your margin is my opportunity." Meaning: While a competitor is extracting profit from some product, Amazon will sell the product cheaper, which either means stealing market share (if that competitor stubbornly maintains its price) or destroying the competitor's profit (if that competitor matches Amazon's price).

But why doesn't this run into the problems we just outlined? Why is this smart for Amazon, but dumb in our hypothetical commoditized-market example?

LOW PRICE AS A COMPLETE STRATEGY

The difference is that successful low-price strategies treat price as **just one component of a comprehensive strategy to win**. A strategy with unique, interlocking, self-reinforcing decisions, where "lowest price" was an *outcome* of the *decisions*, not a tactic thrust upon them by the competition, not their last-ditch effort to win a sale.

Those decisions created **negative consequences** also, that **many potential customers hate**. Great strategy requires strong decisions, which means trade-offs. They created greatness in some areas *by leveraging specific weaknesses* in others. In doing so, they created a strategy that others wouldn't (and didn't) copy, not because no one wanted

to copy their strengths, but because they weren't willing to copy their weaknesses, which enabled those strengths.

These unique trade-offs resulted in a unique product, and although some consumers hate it, many love it. For some, it is the ideal product; for 100x more, it is not ideal, but it is a better set of trade-offs than the alternative. When you dare to trade off, you dare to sell to a subset of the market, but then you win, and you win more than the small "ideal" subset (p. 317) that you targeted.

Let's examine some examples of interlocking, weakness-leading-to-special-strength low-price strategies.

Costco

credit[454]

By coupling higher quality products together with higher quantities in each purchasable package, Costco's *unit price* is lower than grocery stores. Consumers are thrilled to get a deal for something "this good." A store that caters to bulk-buying can also get away with a "warehouse" feel, which means spending less on the interior. Bulk-buying also means they don't need to stock all the goods that a daily grocery store must; this gives them inventory flexibility. These lower costs yield lower prices while also yielding profit. Because other grocery stores can't match these qualities, Costco stores remain unique.

Notice the drawbacks: Unappealing interior, limited scope of inventory, and on top of that, they require an annual membership fee! The fee, by the way, is pure profit, and increases how often a consumer comes back to the store.

Furthermore, to maintain their brand as the low-cost leader, the store has a strict 14% margin cap on all products. This is a clear financial penalty, especially when customers would be willing to pay more—but again this interlocks with their brand promise.

Defying more industry "wisdom," Costco is well-known for treating employees particularly well, including an average $24/hr wage that is

typically double what grocery and retail pay in the same cities. This decreases turnover and increases productivity, which again contributes to profits and a great customer experience. If only all grocery stores would realize the wisdom of this strategy.

Thus, interlocking decisions about employment, quality, package-sizes, inventory, store-constructions, and membership, yields "low prices" as an outcome.

credit[455]

Southwest Airlines

Southwest Airlines has been a low-fare powerhouse for 50 years, yet remained profitable even during the US terrorists attacks of September 2001 and the recession of 2008. Every other major US airline filed for bankruptcy at least once, but not Southwest. Yet Southwest consistently has extremely cheap tickets.

Like Costco, they made interlocking product and operational decisions that generated this result. They shuttle back and forth many times per day along the same few, short routes. Already there's a negative trade-off—no long hauls, no international—but also positive ones—frequent flights means consumers have better schedules and it's easier to "catch the next flight" if needed. They don't interconnect with other airlines, won't transfer your bags, and they fly into secondary airports where major carriers don't operate—more negative trade-offs, but secondary airports are cheaper, so fares can be lower. Other airlines have many kinds of airplanes to support many kinds of routes, but Southwest uses only one type of airplane; this gives them pricing power over the supplier (due to large orders), and is cheaper to maintain (mechanics need only a single set of tools, single set of spare parts, and single training program), reducing costs again. They have no amenities—no 1st-class seating, no meals, no support for travel agencies—which again is a negative trade-off for some consumers, but lowers costs and therefore supports lower prices.

This strategy was so effective—and so well-studied—that it gave rise to a whole sub-category—the so-called "low-cost carriers" that includes JetBlue, Spirit, and Frontier. None were as successful, two of the three

went bankrupt, and none impinged on Southwest's profits, demonstrating that great strategy works even in the face of competition that is expressly "copying you" to replicate your success.

Vanguard Funds

credit[456]

Every large mutual fund conglomerate in the late 1970s included hefty operational fees that paid the salaries of fund managers and analysts who researched stocks and decided when to buy and sell. Consumers paid around 2% per year to have these managers vigilantly generating insights and taking actions on their behalf. If the fund increased in value by 5%, the consumer would receive only a 3% increase. It really hurts if the fund decreases in value. Most funds still operate this way today; even VC and PE funds typically have a 2% annual management fee.

Vanguard realized that many people might want a completely different product: One that simply tracks indexes like the S&P 500, or automatically (i.e. without human judgment) tracks well-defined sets of stocks like "Large American companies." Without teams of human beings, they could eliminate the 2% management fees and expense ratios, replacing it with *de minimus* fees from automatic trades. For consumers who believe that fund managers outperform the market, this new product was silly. But for consumers who believe that managers don't out-perform the market in the long run—especially after removing the compounding effect of fees—the Vanguard funds were uniquely low-cost. Again, "low cost" was an outcome of a unique product, that made different trade-offs, that was appealing only to a subset of the market.

IKEA

credit[457]

IKEA revolutionized furniture retail with a comprehensive strategy where every decision reinforces low prices. The foundation: Forcing customers to assemble their own furniture. Often

poorly, frustrated, and complaining.

This seemingly simple decision created a cascade of advantages—flat-pack furniture dramatically reduces shipping and storage costs, allows for more efficient store layouts, and minimizes damage during transport. Their unique showroom-to-warehouse flow, where customers navigate a single path through inspirational room setups before collecting their own items, both enhances the shopping experience and reduces staffing costs. The globally standardized product line means they're not creating custom products for each market. These decisions come with clear trade-offs—furniture that isn't heirloom quality, the notorious assembly experience, limited selection within categories—but for price-conscious consumers furnishing their first apartment or dorm room or spare bedroom, IKEA's system delivers remarkable value. The low price isn't a tactic; it's the natural outcome of a thoroughly designed system.

LOW-COST STRATEGIES DON'T ALWAYS REQUIRE SIGNIFICANT FINANCING

The modern SaaS playbook dictates that "low price" strategies require raising tens or hundreds of millions in venture capital. The playbook is logical: Low-price business models are profitable only at scale—when revenue finally outgrows the overhead of people, suppliers start providing bulk discounts, sales and marketing costs amortize over enough customers, and so on.

Logical though it may be, it is not the only playbook. All four of these iconic, multi-billion-dollar low-price leaders thrived with remarkably modest initial financing.

Costco's first warehouse opened in 1983 at a cost of $5M, and went public in 1985 with a small $30M offering. Costco's model was profitable almost immediately—the membership fee revenue covered

most of the fixed costs, allowing them to operate on razor-thin product-margins from the beginning. Their expansion was largely funded through operational cash flow rather than massive external financing rounds.

Southwest's initial financing was remarkably modest for an airline. Founded in 1967, they started with $560,000 in initial capital. They began operations in 1971 with just three aircraft serving three cities. Southwest's ability to turn a profit quickly allowed it to fund most of its expansion through earnings rather than heavy external financing.

Vanguard was created in 1975 with an initial capitalization around $2M. Their unique "client-owned" ownership structure meant that it didn't need to satisfy external investors seeking high returns. The index fund strategy required minimal operational overhead from the beginning—no teams of analysts, no expensive trading operations. Vanguard's growth was organic and didn't require substantial external financing rounds.

IKEA was almost entirely self-funded through operational cash flow. Ingvar Kamprad started with a small loan from his father to launch a mail-order business in 1943. The first furniture showroom opened in 1953, funded entirely from the profits of the mail-order business. IKEA's rapid expansion globally was financed primarily through re-invested profits.

Remember, these are physical companies with materials, supply chains, inventory, and buildings. Software companies should be able to do the same with no more investment than this.

The common themes which allowed these companies to beat competitors for many decades in heavily-contested, commoditized markets, with increasing profits, with little financing, all with the lowest prices:

1. Their strategy including trade-offs that many customers find distasteful or a "deal-breaker." By targeting a subset of the market, accepting that for many customers the trade-offs were wrong, they dominated that subset.
2. The weaknesses in their strategy were critical to preventing copycats, whether from existing competitors or new entrants. Others want to copy strengths, but won't copy weaknesses. Because the weaknesses were mandatory to produce their strengths, they were never replicated.
3. Their business models were profitable after just one store or with few products, not only once at scale.
   (In SaaS, we would say: Profitable unit economics, with GPM above 70%, with a CAC payback period (p. 1377) of less than a year.)
4. They reinvested profits into expansion, not extracted as dividends.
   (This is an area where bootstrapped companies often diverge from funded ones, or founders with ambition to create a larger enterprise.)
5. They innovated in operations—not just in customer-facing products and features—to achieve their unique advantages and cost-effectiveness that enabled profit despite having lower prices.
6. They started with narrower offerings in narrow geographic footprints and expanded (p. 793) gradually.
7. They had a long-term vision and held inviolable values (p. 827), never taking short-term short-cuts or compromises.

This should give hope to bootstrappers everywhere, or to folks wanting to raise sustainably-small rounds from strategic investors and advisors.

When the only differentiation a product can boast is that "it's cheaper," that's not a strategy at all. The result is commoditization and a race to the bottom—if it gets off the ground at all. That's why you mustn't compete *only* on price.

That's being *cheap*, not *affordable*. That's a lack of vision, not a comprehensive strategy.

When lower prices are *a result* of strategy—business structure, cost structure, product trade-offs (that ICPs (p. 317) love but others hate), and other decisions that competitors can't or won't make, then low prices are a powerful, winning strategy.

Chapter 26:
Business Advice Plagued by Survivor Bias

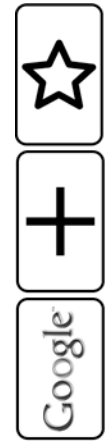
BULLET HOLES · BURYING EVIDENCE · ESP
EXAMPLES · OOPS

Do you read articles written by a founder who failed three times, never finding success?

No, because you want to learn from success, not hear about "lessons learned" from someone who hasn't yet learned those lessons themself.

However, **the fact that you are learning only from success is a deeper problem than you imagine.**

Some stories will expose the enormity of this fallacy.

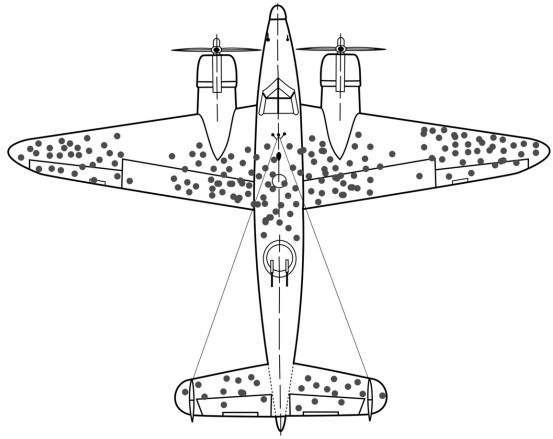


BULLET HOLES: A BRAIN TEASER

During World War II the English sent daily bombing raids into Germany. Many planes never returned; those that did were often riddled with bullet holes from anti-air machine guns and German fighters.

Wanting to improve the odds of getting a crew home alive, English engineers studied the locations of the bullet holes. Where the planes were hit most, they reasoned, is where they should attach heavy armor plating. Sure enough, a pattern emerged: Bullets clustered on the wings, tail, and rear gunner's station. Few bullets were found in the main cockpit or fuel tanks.

The logical conclusion is that they should add armor plating to the spots that get hit most often by bullets. But that's wrong.



credit[458]

(Note: This oft-sited image is not the actual data[459] from the original report.[460])

Planes with bullets in the cockpit or fuel tanks *didn't make it home*; the bullet holes in returning planes were "found" in places that were by definition relatively benign. The real data is in the planes that were shot down, not the ones that survived.

This is a literal example of "survivor bias"—drawing conclusions only from data that is available or convenient and thus systematically biasing your results.

Doesn't most business advice suffer from this fallacy? You read about successes but what about the businesses that "never made it home?" Like the downed planes, **could failure contain more lessons than success?**

BURYING THE OTHER EVIDENCE

Scientific journals publish extraordinary results, so studies whose results are statistically insignificance aren't published. Rather, they are abandoned or silently stowed away in academic filing cabinets.

For this reason, this practice is called the "file-drawer effect," and it's a particularly insidious form of survivor bias because it is invisible. Peter Norvig sums it up nicely:[461]

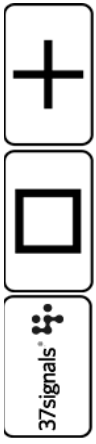
> When a published paper proclaims "statistically, this could only happen by chance one in twenty times," it is quite possible that similar experiments have been performed twenty times, but have not been published.

Pharmaceutical companies have exploited this effect to skew results intentionally. It's gotten so bad that journals are calling for a public database to prevent fraud:

> More than two-thirds of studies of anti-depressants given to depressed children, for instance, found the medications were no better than sugar pills, but companies published only the positive trials.
>
> If all the studies had been registered from the start, doctors would have learned that the positive data were only a fraction of the total.
> –Washington Post[462]

Doesn't most business advice suffer from this fallacy? Harvard Business School's famous case studies include only success stories. To paraphrase Norvig, **what if twenty other coffee shops had the same ideas, same product, and same dedication as Starbucks, but failed?** How does that affect what we can learn from Starbucks's success?

EXPERIMENTAL PROOF OF ESP



Dr. Joseph Rhine[463] brought the rigor of experimental psychology to the study of the paranormal, and ESP (Extra Sensory Perception) in particular. He made waves in the 1930s with controlled experiments testing whether a person was able to predict the order of the cards in a shuffled Zener[464] deck (with symbols like circle, square, star, and wavy lines).

In a typical experiment, 500 people are screened for "strong telepathic ability," measured by significantly above-average performance in a 25-card deck. Those selected are tested again; most drop away. Tested a third time, perhaps one person passes again and we conclude that such a repeat performance is statistical evidence of genuine ESP.

To see why this is just a different face of survivor bias, consider the following experiment. I believe some people are "heady" when it comes to coin-flipping—getting heads more often than chance alone would suggest. So I put 1000 people in a room and tell them to flip a coin ten times. Sure enough, a woman named Margaret makes "heads" ten times in a row! The chance of her getting heads ten times in a row is only 1-in-1024, so I conclude Margret has special abilities.

Actually that last statement is true but misleading. The chance that *Margaret* would flip ten heads in a row is 1-in-1024, but that wasn't the experiment I ran, was it? I let 1000 people flip and "found" Margaret in the crowd.

The chance that *somebody* in a crowd of a thousand would flip heads ten times is a whopping 62%! Because so many people are attempting the feat, some normally-unlikely events will happen. This isn't a test of Margaret's abilities at all!

Doesn't most business advice suffer from this fallacy? Take me for instance. I've started three* consecutive successful companies, so that's proof that I know what I'm doing and that you should do everything I say, right? Except maybe I'm just the one in the crowd who guessed right on the Zener cards three times, and there's no reason to believe I would be successful a fourth time.**

---

* Author's note in 2023: Now four with WP Engine;[465] with 200,000 customers and hundreds of millions in ARR, and profitable, it's an uber-Margaret!

** Author's Note in 2023: Looking back fourteen years, having done it again, I still wouldn't change a word. Maybe I'm still Margaret. Perhaps I've earned additional benefit of the doubt, but the survivor bias warning remains.

SPECIFIC EXAMPLES OF SURVIVOR BIAS IN BUSINESS ADVICE

So far I've been asking rhetorically whether survivor bias *might* be severely skewing business advice. Steven Levitt[466] (of Freakonomics[467] fame) investigated this question directly.

He was reading Good to Great[468] by Jim Collins, a book that analyzed eleven companies that were mediocre—just pooping along—but then transformed themselves into stock market sensations. A conclusion was that the common trait was a "culture of discipline." This book has sold many millions of copies, so it's a good example of popular writing on business advice.

One of the eleven "great" companies was Fannie Mae, and Steven Levitt was reading this book just as Fannie was collapsing in financial disaster. Hmm, he thought, I wonder how those other "great" companies are doing.

Turns out, had you invested in those eleven companies in 2001 (when the book came out), your portfolio would have underperformed the S&P 500! (Fannie Mae wasn't even the only case of total disaster—also extolled was the now-bankrupt Circuit City.)

Why didn't these companies continue to succeed? It turns out Jim started by combing through 1435 companies looking for good candidates for the book, and picked eleven. **It's the ESP experiment all over again!**

On top of that, Jim doesn't bother asking whether any of the 1424 other companies also displayed a "culture of discipline." Maybe that's something that many public companies have regardless of performance.

Is this book an aberration? Nope, Steven investigated another business book from the 1980s—In Search of Excellence[469]—and found the same effect.

Steven then comes to the same conclusion that I'm coming to:

> These business books are mostly backward-looking: what have companies done that has made them successful? The future is always hard to predict, and understanding the past is valuable; on the other hand, the implicit message of these business books is that the principles that these companies use not only have made them good in the past, but position them for continued success.

To the extent that this doesn't actually turn out to be true, it calls into question the basic premise of these books,* doesn't it?

OOPS, DID I JUST INVALIDATE THIS ENTIRE WEBSITE?

Lately I've been wondering if a lot of business advice—both mine and others—is really a case of survivor bias. I mean, I didn't start out at Smart Bear with a load of philosophy and a fixed idea of who the customer was or even what the products would be.

How do I know this post-hoc philosophy and advice isn't just a case of survivor bias? Am I not like the ESP-savant, successful not by force of nature but by simple chance of surviving?

Or perhaps I'm like Dr. Rhine the ESP experimenter—convinced I've discovered something important with "objective measures of

* I personally love *Good to Great* anyway, because I think the ideas in there are timeless and excellent.

success"—and yet I'm actually living in an egotistical, possibly even narcissistic dream world.

More to the point, **how can *you*, dear reader, ascertain whether my articles or any advice from anywhere suffers from this fallacy?**

In the end of course you *don't* know. But here's something: Just the fact that you're aware of survivor bias means you're less likely to be taken in by it. So, reading this article has helped a little.

Beyond that, prefer advice that makes you think rather than giving you answers, forces you to answer tough questions, and causes you to extend your existing strengths (p. 543), and become a better version of the person you already are (p. 751).

Chapter 27:
Impostor Syndrome: Why I felt like a fraud, and how I overcame it

"**I feel like a fraud.** I've been at this for 16 years and I still feel like a fraud. I'm just waiting for the day they see through the façade, but they keep coming back every year."
—Jason Young

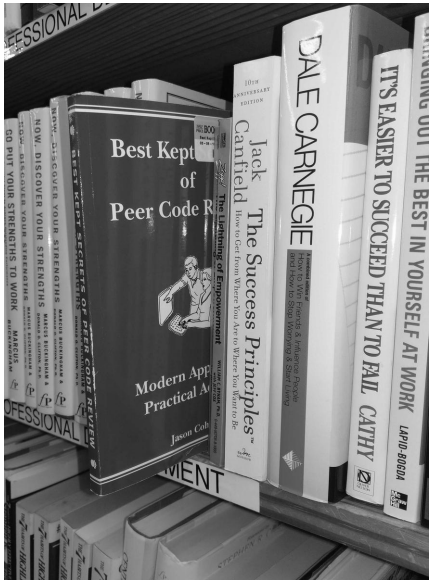
Ah yes, the awe-inspiring words of confidence from the seasoned entrepreneur. My friend Jason intended this as soothing words of solace during (one of my) periods of personal freak-out while Smart Bear was in its infancy.

I felt like a fraud every day. Here I was, selling a wobbly, buggy tool and pawning myself off as an expert in a field that didn't exist.* Every second I felt like I was putting one over on the world.

I would explain how our tool "cuts code review time in half," but was that actually true or had I just repeated the argument so many times that I stopped questioning it? I would orchestrate purchases, but should I be handling large sums of money with no knowledge of accounting,[470] cash-flow, invoicing, purchase orders,[471] all while being relentlessly belittled by Accounts Payable (p. 737)? I would instruct customers on "best practices" for code review, even publishing a whole book about it, but who am I to tell other people how to critique code?

Aren't I too young? Isn't the tool too crappy to charge for? Aren't I too inexperienced? Don't I need an MBA or at least some sales training?

Is Smart Bear a "real company?" What does that even mean (p. 1113)?



* Our software was the first commercial tool for peer code review. Now a standard practice for many developers, and a standard feature in many development tools, at the time we were creating a market that didn't exist.

Objectively, and with hindsight, my feelings were misplaced. The tool really did save time and headache; customers said so. As much as I doubted the title "Code Review Expert," I had developed more experience with more teams in more situations than any one person could (because everyone else was busy doing their actual jobs). And sales isn't as mystical and unknowable as I feared.

Still, **emotions don't respond to logic.** Jason was telling me that these feelings don't go away, even when they ought.

The other thing he was saying is: **You're not alone.** It's not just business founders. Mike Meyers said "I still believe that at any time the No-Talent Police will come and arrest me." Jodie Foster said "I thought it [winning the Oscar] was a fluke. The same way as when I walked on the campus at Yale. I thought everybody would find out, and they'd take the Oscar back." June Huh, who dropped out of high-school to become a poet, and later won the Fields Medal—mathematics' highest honor—says[472] "Of course you are happy, but deep down, you're a little bit worried that they might eventually figure out that you're not actually that good. I am a reasonably good mathematician, but am I Fields Medal-worthy?"

The name for this is Impostor Syndrome.[474] Studies show[475] that "40% of successful people consider themselves frauds" and that "A staggering 84%[476] of the entrepreneurs and small business owners surveyed had scores indicating they feel like an imposter at moderate, frequent, or intense levels." Ask any small business coach; they'll confirm how prevalent these feelings are.[477] It's also common with PhD candidates.[478] People constantly complain of it on Twitter,[479] and that's just the <1% who are (a) on Twitter and (b) willing to admit it.

Although not an official psychological disorder, and generally not crippling, if you have these feelings it's useful to know that it's common and there's something you can do about it.

See if these sound familiar:

- You dismiss compliments, awards, and positive reinforcement as "no big deal."

credit[473]

"I think instead of worrying about why people don't believe in you, we should worry about why you don't believe in *yourself*."

- You are crushed by mild, constructive criticism.
- You believe you're not as smart/talented/capable as other people think you are.
- You worry others will discover you're not as smart/talented/capable as they think you are.
- You think other people with similar jobs are more "adult" than you are, and they "have their shit together" while you flounder around.
- You feel your successes are due more to luck than ability; with your failures it's the other way around.
- You find it difficult to take credit for your accomplishments.
- You feel that you're the living embodiment of "fake it until you make it."

But wait, how can this be? **This overwhelming lack of self-confidence is the opposite of the traditional entrepreneurial stereotype.** Don't founders forge ahead even when others say success is impossible? Doesn't a founder invent a new product based on her confidence that others will want it (p. 71)? Doesn't the very idea of

starting your own company scream "I'm doing it my way, and my way is better?"

Here's why it makes sense. Consider what it means to be a perfectionist. The perfectionist sees flaws in everyone else's work; there's always a way to make it better—*her* way. She doesn't respond well to authority dictating how things must be; neither is she comfortable delegating (p. 981) to those who (by her definition) clearly don't care as much as she does.

Sounds like the stereotypical attitude of the arrogant startup founder, but wait! At the same time, the perfectionist is never happy with her *own* work either, seeing (inventing?) a never-ending stream of flaws that require attention. No matter how highly others regard her work, the perfectionist insists it's incomplete and unsatisfactory. She can't accept the idea that others would be impressed with her accomplishments, because to her eyes they're mediocre works-in-progress. She worries that one day they'll realize she's right.

Our entrepreneurial motivation is not confidence, it's an insatiable desire to do it our way. Doing it our own way is not the same as being confident that it is the correct way. It's about never accepting any idea as being best, even if you like your current ideas better than everyone else's.

Can these feelings be constructive? Yes, if they're a sign that you're striving to learn and improve. As Andy Wibbels says:

> If I don't feel like a fraud at least once a day then I'm not reaching far enough.
>
> If you aren't scared shitless then why bother?

Here's what it looks like when you're channeling these self-doubts into something **constructive**:

- I doubt my title as "expert," so every day I read, write, and immerse myself in my field.

- I doubt the quality of my software, so I fix bugs as fast as possible, I write unit tests proactively, and I thank my customers for their patience.
- I doubt I deserve my reputation, so I work hard to earn it.
- I'm not as good as I want to be at speaking/writing/programming/designing/managing, but I can see myself slowly improving.
- I'm not a "real company (p. 1113)" yet, so I concentrate on making my customers successful, so they don't care about corporate size or structure.

On the other hand, here's what it looks like when these doubts are **harming** you:

- I doubt my title as "expert," so every night I worry about what will happen when I'm discovered as a fraud. I'm absent-mindedly looking for trivially-easy jobs I could take where this pressure won't exist. (Looking for an "escape-hatch" is a well-documented behavior.)
- I doubt the quality of my software, so I spend lots of time covering it up with graphic design and heavy sales pitches.
- I doubt I deserve my reputation, so I live in constant fear of exposure. I can't sleep at night and I loathe myself for lying.
- I'm not as good as I want to be at speaking/writing/programming/designing/managing, so I go out of my way to avoid any of it, and feel like a trapped animal when I'm forced to do it.
- I'm not a "real company" yet, so I feel guilty every time someone gives me money or believes anything I say.

If you're letting these feelings get to you too, at least recognize it so you can deal with it logically.

"Turns out my imposter syndrome was pretty much right on."

credit[480]

Here's some specific ways to answer the Impostor voice in your head:

...yet
Whatever it is you think you're not good enough for, append the word "yet." I can't do this job... yet. This article isn't good enough... yet. I don't deserve the title I have... yet.

"Am I a curious student?"*
A: When you meet a student, who is honesty, energetically, and humbly pursuing learning and growth, do you think "That's awesome, good for them, how can I help," or do you think "They're an impostor for not already being an expert?" So, when you act as a curious student, that's not being an impostor, that's the state of being a student. Which is awesome.

* Inspired by Mihir Patel's article[481]

"Do I deserve to be peers with my peers?"
A: Yes, unless you think Neil Gaiman doesn't deserve to be at the same party[482] as Neil Armstrong or vice versa, because they feel the same way, and you know they're both wrong.

"Do I deserve the [title/position/responsibility] I have?"
A: Do you think some idiot gave you a hand-out for nothing? If not, then yes, you deserve it.

"Sometimes I feel stretched and that I don't know what I'm doing."
A: That is the human condition... for humans who strive. Only someone who is ignorant and foolhardy doesn't realize there's more to know, more to learn, and to be suspicious of confidence. If Mike Meyers isn't sure whether he's funny, then it's just the human condition. Even if sometimes you *are* stretched too far, that's *also* the normal condition when striving. It's something to embrace and work on, not evidence that you shouldn't be doing it at all. Unless you think "learning and growing" is something you should never do, or if you think "learning and growing" should never feel uncomfortable.

And when logic fails, maybe this will help:
You believe that Mike Meyers and Jodie Foster are talented, right? And June Huh deserves that medal? You might even believe that I'm an expert in peer code review and startups in general. Yet we doubt ourselves every day. *And we're wrong.*

So if we're wrong, you're wrong too.

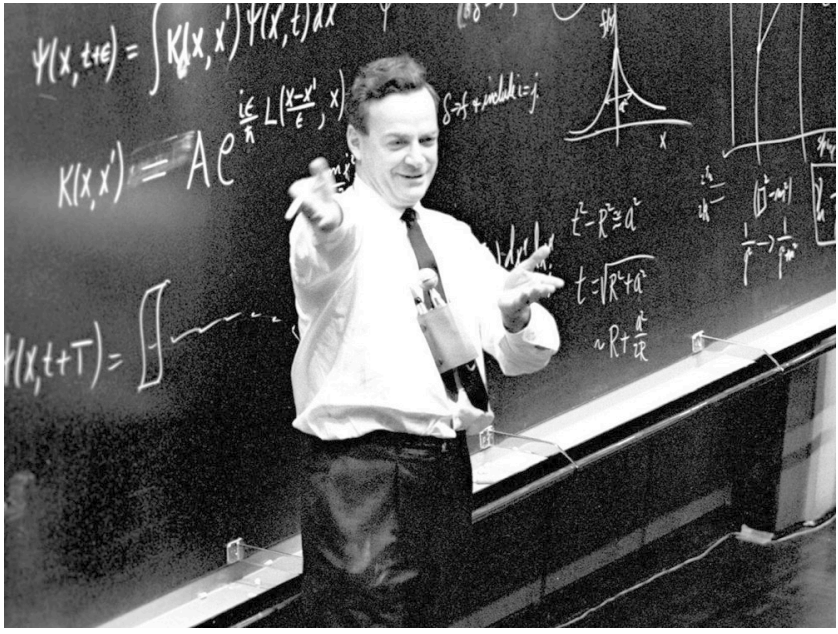
Don't stop striving, just stop holding yourself to an impossible standard, and don't worry what other people think.

They're not thinking about you at all.

Many thanks to Matt Cohen[483] and Scott W. Bradley[484] for contributing examples.

Chapter 28:
All pretty models are wrong, but some ugly models are useful

WRONG, BUT USEFUL · BEAUTIFUL, BUT WRONG



“Study hard what interests you the most in the most undisciplined, irreverent and original manner possible!” —Richard Feynman

WRONG, BUT USEFUL

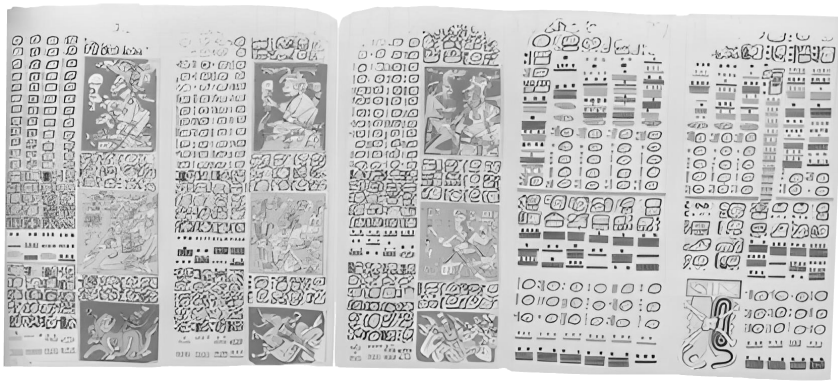
Contrary to our intuition, a theory of physics can be **useful**—making accurate predictions—and yet simultaneously **incorrect**—not accurately describing how the universe actually works.

The Mayans' conception of the Earth, moon, sun, planets, and stars, was as ludicrous as every other ancient civilization, yet their priests routinely predicted the timing of eclipses with impressive accuracy. The priests leveraged this accuracy as evidence that their religion was correct, and that their exalted position in society was justified.

Their religion—and therefore their explanation of how the universe worked—is laughable to the modern reader: The Earth in the center (of course), with thirteen tiers of heaven whirling above and nine levels of underworld threatening from below. Eclipses are not caused by a physical object blocking the light of the sun, but rather spiritual beings temporarily consuming the sun or moon (Figure 1). Even the most fervently religious person today would classify these ideas as fanciful mythology, though the Mayans were no less certain of the veracity of their religion than modern-day humans are of theirs.

Nevertheless, they were careful observers and meticulous calculators. They understood that eclipses happened roughly every 173 days, adjusted by a 405-month cycle and additional smaller correction. They tracked these cycles and updated their model over the centuries, and as a result, **their theory yielded accurate predictions, even though the theory's explanation of *why* was entirely incorrect**.

This is a striking example of the common bromide: All models are wrong, but some models are useful. Their model was *wrong* in describing the universe, but *useful* in predicting eclipses.



credit[485]

Figure 1: Recolorized segment of the Dresden Codex,[486] a Mayan text from around 1200 AD which, among other things, predicts eclipses to within a few days.

A thousand years later, modern physicists still wrestle with the question of whether their models explain how the universe actually works, or are merely useful. Physicist Richard Feynman illustrated this point using the Mayan example above. Quantum Mechanics (QM) is so weird and counter-intuitive, Feynman says, that "nobody understands quantum mechanics."* And this is from someone who won a Nobel Prize for creating an accurate and useful model for how some of it works.**

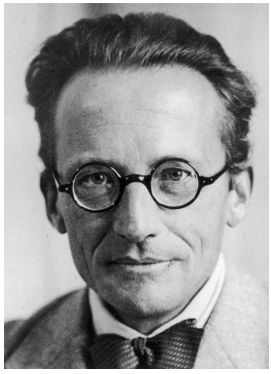
Feynman (jointly) won the prize for modeling how particles interact with each other. The model sounds as crazy as Mayan mythology: It says that *all possible interactions happen simultaneously*** (yes, "everything, all the time"), with every interaction-possibility reinforcing or

* From *The Character of Physical Law*. This is sometimes erroneous quoted as: "If you think you understand quantum mechanics, you don't understand quantum mechanics." That's a cute paraphrase, but he didn't postulate that someone would even *believe* they understood it!

** Electrons and photons specifically, but it turns out to be the right model for all types of subatomic particles.

cancelling-out other possibilities, and with each interaction weighted by the probability of its occurrence. When we make a measurement in the lab, the cosmic dice are thrown, and one of those interactions is observed to have occurred in actual fact.

This, of course, makes no sense. This sounds like Mayan cycles that miraculously spit out the correct answer, not how the universe could really work. Albert Einstein thought as much, famously trolling "God does not play dice with the universe." To prove his point he, along with Boris Podolsky[487] and Nathan Rosen,[488] described the "EPR Paradox"[489]—an experiment where the QM model predicts an even more "absurd, impossible" result, therefore (they felt) "proving" that the QM model is a more sophisticated kind of Mayan cycle computation, and might even be downright incorrect. Unfortunately for Einstein, physicists have run the EPR experiment many times in the subsequent decades, and the model-nobody-understands has always been correct in every detail.



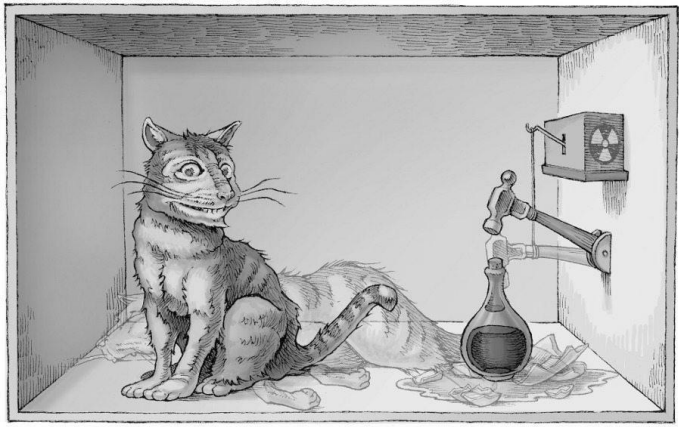
credit[490]

Erwin Schrödinger in 1933

Erwin Schrödinger was also personally entangled with the apparent problem that the QM model was absurd, yet useful. His Schrödinger Equation[491] is the center-piece of QM: It dictates how the entire universe evolves over time. Nearly every QM calculation runs through this equation. And yet, like Einstein, Schrödinger agreed that although the model is successful, its description of how the world works is ludicrous, quipping "I don't like it, and I'm sorry I ever had anything to do with it."**** He invented the famous Schrödinger's Cat[492] (Figure 2)

*** The precise description is: Sum the probability amplitudes and phases over all paths the particles could have taken, including an infinite hierarchy of virtual particle interactions. I ask the pedantic reader to forgive my evocative simplification.

**** Here "it" refers specifically to the Copenhagen interpretation of the collapse of the wave function when an observer observes it; another facet of QM that obviously



credit[493]

Figure 2: The QM model says that Schrödinger's cat is simultaneously alive and dead until you look inside the box; another example of "everything happens, but with probability, resolved when you observe it." Both the "simultaneously" and "when you observe it" are nonsensical concepts (how does the cat-system "know" that some human-system "observed" it?), yet hundreds of experiments have confirmed the predictions of the model.

thought-experiment to prove that QM must not be describing how the universe "actually" works, just as Einstein attempted with EPR. And, like Einstein, his attempt failed; physicists have run this experiment dozens of ways over nearly a hundred years, and the model has always been correct.

makes no sense as soon as you ask "What do you mean by 'observer' and why isn't that just another physical system?"

BEAUTIFUL, BUT WRONG

And so we come to models of companies, markets, and people.

Economics, modeling how companies operate in isolated, simple paradigms (micro) or in bulk (macro). Management theory, modeling how information and control and human behavior flows across organizations. Strategy theory (p. 489), modeling a company's most important constraints and levers, strongest capabilities and assets (p. 569), modeling competitors and the market at large, resulting in the top-level decisions that we hope will bring success. Product Management (p. 817), modeling customers' whims, incentives, "pain-points," "delighters," "JTBD," and willingness-to-pay (p. 275). Startup theory, giving frameworks for methodically transforming an idea into profit (p. 9) while avoiding failure (p. 379).

All of these models work some of the time. All explain the past retroactively but often fail to predict the future, and thus none are on par with theories of physics, and arguably shouldn't be called "theories" at all. A company is not an experiment (p. 913) with controlled variables and many trial-runs. Even using the term "expected value" is a fallacy (p. 963).

Are these models more like the Mayans or more like QM? Are they occasionally useful but not representative of how the world actually works, or do they indeed model how the world operates, where their outputs are incorrect not because they are intrinsically flawed, but because of noisy environments, faulty inputs, missing inputs, or human operators who are more interested in selling consulting services off their own HBR articles and books than they are in acknowledging the limits of our simplistic models in the presence of an irreducibly complex world?

This is partly how I judge frameworks: Does the framework attempt to model the messy real world, or does it seem like a pretty fantasy that looks nice in presentations, HBR articles, and consultants' sales materials?

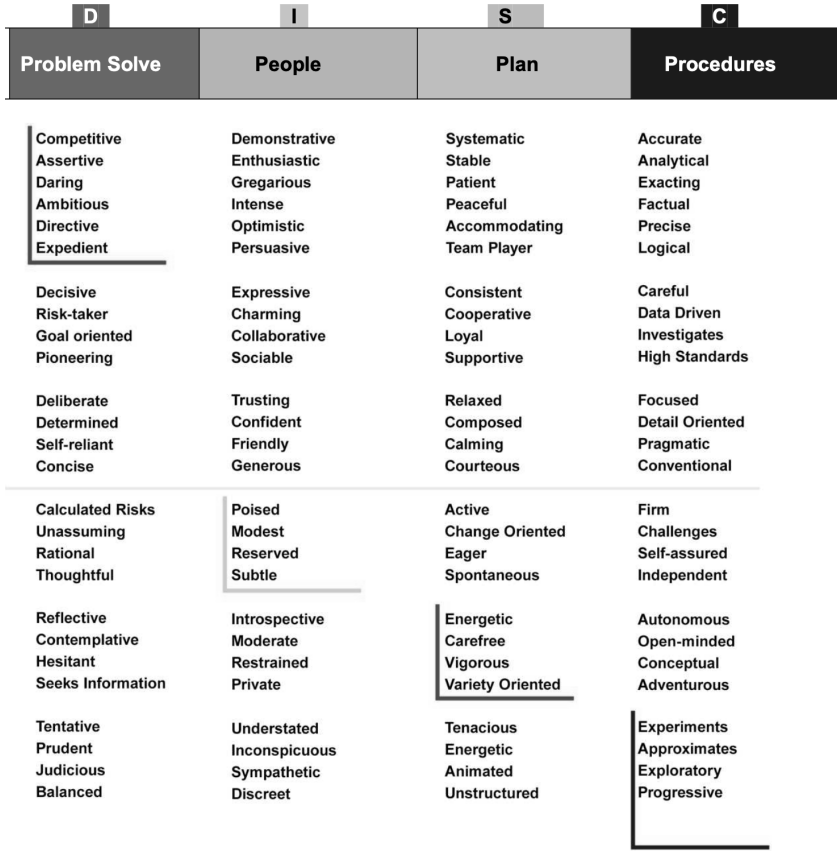


Figure 3: The DISC personality assessment

For example, I immediately disbelieve any framework about human beings that comes packaged in a nice, symmetric diagram, with identical quantities of bullets and sub-categories (Figure 3).

Human beings are more complex than saying "there are four categories, and each of the four have exactly six subcategories of descriptors, and each of *those* have an identical number of components." No, that's never how it is with people. If you had said there are five major categories, and some don't subdivide, while others are complex, and some are fairly well-understood, while others are still a mystery, I'd

Figure 4: Craigslist listings for Austin, TX

believe that you were trying to model reality instead of ensuring some picture had 90° rotational symmetry.

A more accurate model of human activity is seen in Craigslist, where they have "however many" categories containing "however many" subcategories (Figure 4).

Human emotion had better *not* be modeled as a perfectly-symmetrical list. In 1980 Dr. Robert Plutchik created a better model, designed to help his patients identify their feelings—quite useful! It is suspicious in that every second-level category contains exactly two third-level emotions, but at least the first-level categories vary in quantity (Figure 5).

Another red flag is any list with exactly 10 items (p. 1249). That's how many fingers we have, not how many things should probably be on that list. Instead, my lists of things like what goes into a great strat-

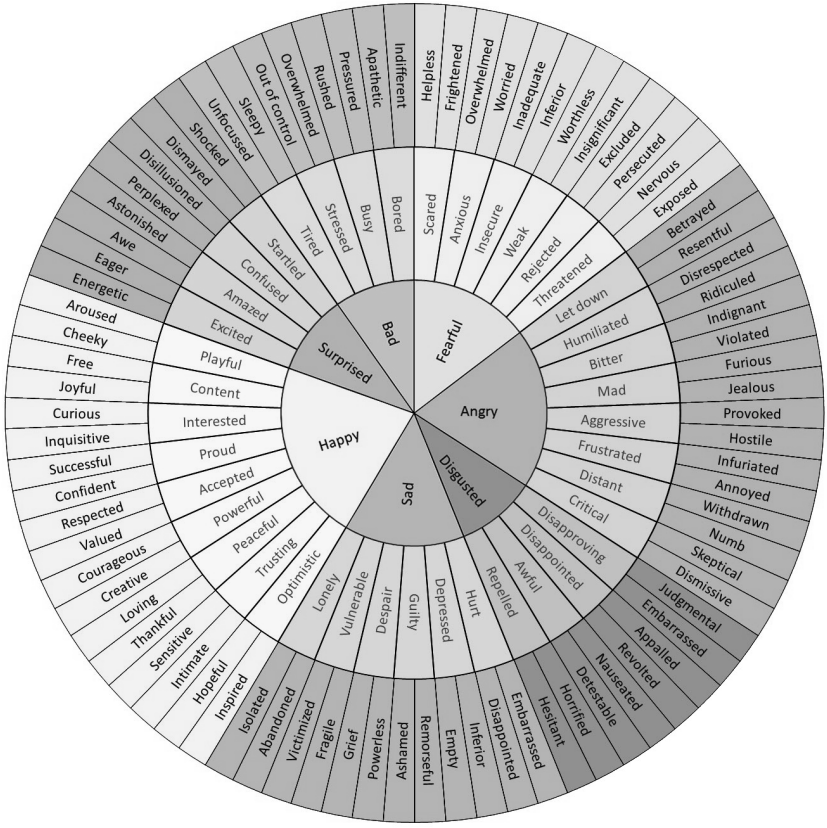


Figure 5: "Feelings Wheel"
Plutchik, Robert. "A General Psychoevolutionary Theory of Emotion." Theories of Emotion, edited by Robert Plutchik and Henry Kellerman, Academic Press, 1980, pp. 3–33.

egy (p. 489) or deciding whether an investment is worthwhile (p. 867) contain however many items make sense. Or in this analysis of why startups fail (p. 379), the categories and quantity of bullets under each category are imbalanced. Or my system for PMF (p. 9) has steps of varying length and detail, and even gives counter-examples to show how it's an interesting guide but not a law.

And so on with other models. If you're modeling human organization, does it reflect the complexity and capriciousness of real humans,

credit[494]

Figure 6: Obviously "marketing" does not have exactly these quantities of these things, that just so happens to perfectly fit the periodic table of elements.

the good and bad incentives and emotions, the different personalities and ways those react to the world, or is it modeled as if people are fungible worker-units with predictable responses to stimuli, built by a professor who has never managed a team beyond six sycophantic grad students?

Does the model of strategy fit into 2x2s and symmetric diagrams, with rubric scoring, with the same questions for all companies of all stages in all industries in all markets? Or does it grapple with the complexity of interacting systems of markets, customers, competitors, alternatives, employees, technology, products, and global trends, each of them dynamic, each affecting the others, each unknowable and unquantifiable in some of their most important dimensions?

I trust more in diagrams that are imbalanced, asymmetrical, or even ugly, because perhaps they're primarily interested in modeling the messy truth of the world, like in Figure 7 and Figure 8.

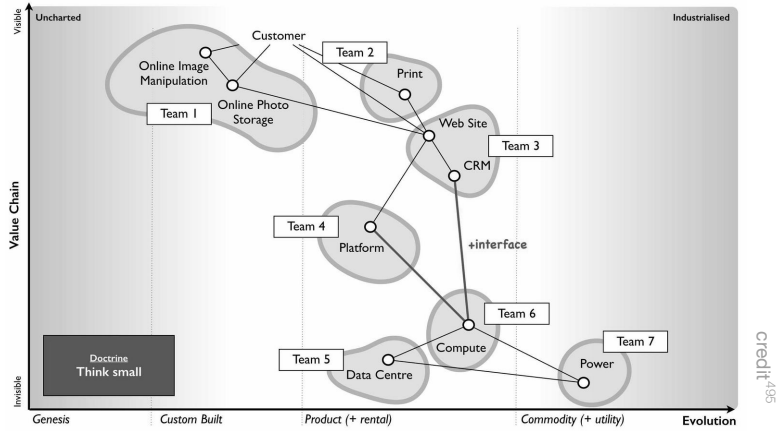
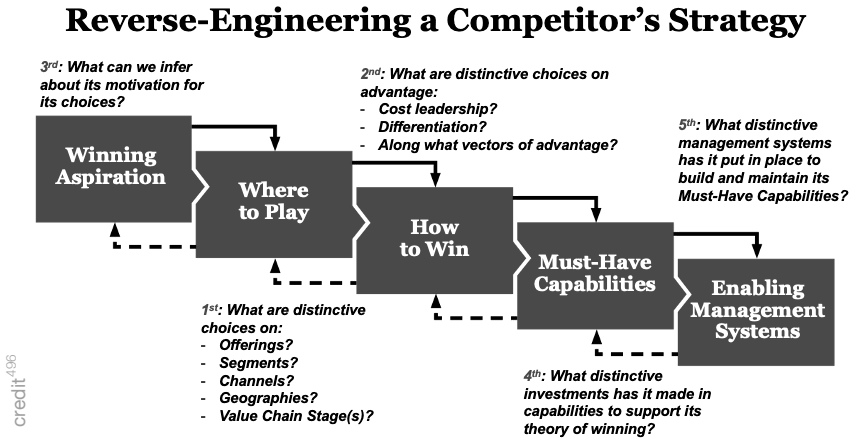


Figure 7: Wardley diagram, sussing out which activities support growth and differentiation, versus which are necessary but commoditized, so that teams can respond with appropriate investment and action.

All models are wrong, but some are useful. The most useful are the ones that genuinely attempt to model the real, complex, ugly, asymmetric world, not the ones made to look pretty on brochures for consulting services.



credit[496]

Figure 8: Though Roger L. Martin's slide has "five boxes," the instructions have "however many questions are useful," and the order-of-operations is non-linear.

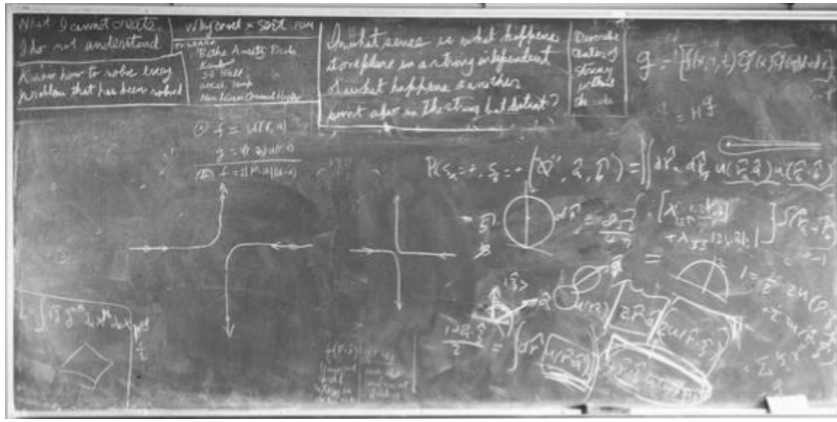
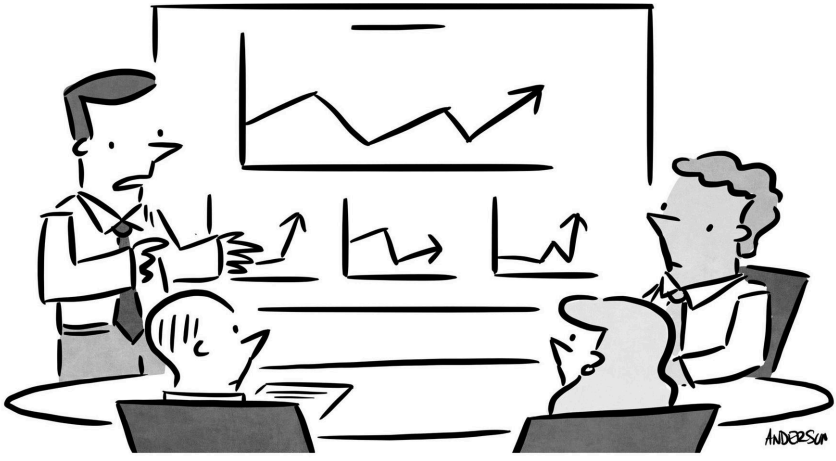


Figure 9: Feynmann's blackboard when he died. The boxed statements at top-left are:
"What I cannot create, I do not understand" and
"Know how to solve every problem that has been solved"

Chapter 29:
Product Purgatory: When they love it but still don't buy

MAGIC WAND · TODAY IS THE DAY



"Executives who like this sales graph also liked these."

credit 497

Most people say nice things when you pitch your product. "Oh, that looks really great! I like it, but we're not buying right now. Maybe call back later in the year?" It's hard to tell you that your baby is ugly.

This creates a trap for new founders who are (rightly!) infatuated with their ideas. They yearn so desperately for validation—nominally product validation but actually personal validation—that they interpret these kindly social white lies as confirmation.

But sometimes the potential customer *really does* love it. They *really do* wish they could buy it. And yet, they're still not buying, or at least "not yet." You have the distinct feeling they weren't just being nice—they were genuine. And yet, the purchase never happens.

This is Product Purgatory.

NOT EVEN WITH A MAGIC WAND

It's amazing how many products fail an even simpler test of desirability and viability. I call it the Magic Wand Test, and it works like this:

> I have a magic wand, and when I wave it, our product will be fully implemented in your company. Everyone is trained. Other tools are integrated. Processes are changed. Security requirements are satisfied. And best of all, it is *completely free*!
>
> So! Do you want me to wave this magic wand?

Shockingly, the answer is often: **"No."**

How could they say "No!?" If the product delivers any value whatsoever, you might as well take it, right? No, because all products not only create friction in implementation, but risk and time and money, often on-going, and the Magic Wand Test proves it. The value of the product must *significantly exceed* this activation energy.

A great example of this was an Austin-based startup twenty years ago. There were a spate of news stories about employees leaking secret corporate documents via email.* So this startup created a solution: All corporate email would flow through the startup's servers, where the content is scanned for sensitive material. Questionable things could be returned to the sender or otherwise flagged.**

Despite the very real problem they solved, they couldn't get anyone to buy it. The reason wasn't that people didn't care about leaks; the news stories were real. The reason was that, even if you could have it for free, there is just too much risk in putting a brand new startup in the middle of all your email. There's the obvious security issue of a probably-hackable startup, and someone else's employees potentially having access to all your email. Also bugs. If this new company has

---

* This risk continues to the present day. In 2017 a Boeing employee accidentally emailed his spouse[498] a spreadsheet with personal information of 36,000 coworkers. Boeing offering two years of free credit monitoring to affected employees at an estimated cost of $7,000,000.

** In another fascinating example of how companies solve this problem on their own, in 2008 Tesla faced a situation where confidential company information was being leaked to the press. Tesla implemented a clever method[499] to identify the leaker: Send a unique email to each employee, differing only in things like white space and punctuation, so that when the press reprinted that information using copy/paste, those details would identify the source.

bugs—and doesn't all new software have bugs?—that would compromise email, which is catastrophic. So although there was some value, and a real problem, the value did not significantly exceed the penalty of using the product.

In the real world the Magic Wand doesn't exist; even if the price is fair, there are real costs and disruption of implementation and training and integration, and people simply getting used to a new user interface and processes and therefore being unproductive for a while. Product inventors revel in all the value they generate, but don't see the penalties that they incur. The customers, however, see.

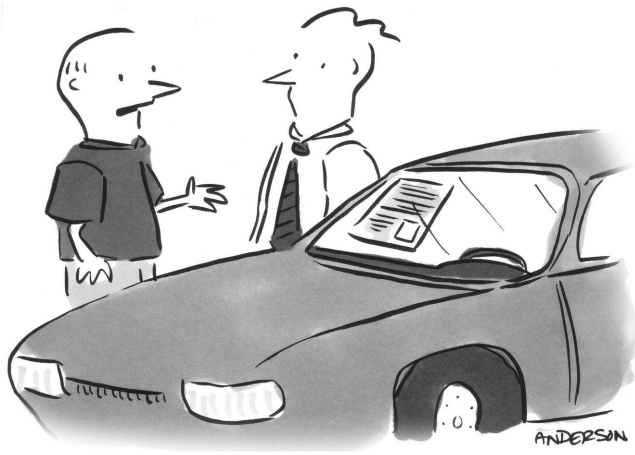
Furthermore, it's not good enough for value to merely exceed penalty; the value must be *far larger* than the penalty. If the value is only slightly larger, it's not worth engaging, not worth the money, not worth the risk. Change is difficult (p. 1299) and time is limited (p. 725); customers cannot spend their time on things that have only minor impact.

This challenge looms even larger for new companies, because you have neither brand nor track record. That's even more risk, including that you're out of business in two years or that your product can't handle real-world use-cases. Even for free, many types of customer are not willing to take those risks. I don't want to use "the newest code library" for anything except fun throw-away projects, because I don't want to rip and replace in a few years, and I want thousands of bugs to have already been found and eradicated, and I want to be able to hire people who already have experience with the library.

It's surprisingly hard to pass the Magic Wand Test, and this is sometimes the answer to the original puzzle of why the customer appears to like it but does not buy.

But now let's go further and consider the product that really does pass the Magic Wand test, but still isn't bought.

What else is missing?



"I like it, but I'm looking for more of a status symbol. Any way you can double the price?"

credit500

MOTIVATION & URGENCY: WHY IS TODAY THE DAY?

The sales call goes well. They're engaged, asking great questions. They explain how they would use this in real life, how they would integrate this into their processes, how it would save them time or save them money (p. 165) or create new opportunities for them. They think the price is fair. They don't muse wistfully about competitors, and they hate what they currently use.

In short, your product clearly passes the Magic Wand Test. It's not your imagination or desperation seeing interest where there is none. And yet... they still don't buy.

Because they're genuinely interested, they give you genuine excuses: "We can't implement this right now but call back in nine months." This excuse points to the answer.

At all times the person with buying power has one, two, or three top-priority things on their plate. These are the things in their quarterly plan (p. 1065), or there is a pressing emergency that has overridden their plan. They're working on those things currently, or they're up next.

Your thing is not one of those three priorities.

Maybe you're priority seven or ten. You're on the list—that's what's different between this case and failing the Magic Wand Test where you're not even on the list.

They won't displace a top-three priority for you. Fair enough; you don't expect them to. And since you're not number four either, you're not coming up in the immediate future. This is why they're putting you off for a year—by then, perhaps they've addressed the top-priority things of the moment, so they might have time for you.

This is the source of Product Purgatory. You got so many things right (p. 71)—the problem exists, the customer knows it, the customer has budget, the customer has the desire… but there's no urgency, no reason it has to be done now. **Urgency is the missing ingredient.**

A good example of this in the world of website development is "accessibility"—ensuring websites work well for people with diverse abilities and disabilities. This includes support for screen readers, resizable text, semantic navigation, high contrast modes, and alt-tags for images.

Having an accessible website passes the Magic Wand Test. At our company WP Engine, we made both our website and user portal

accessible, which not only generated appreciative messages but also sales, as those customers chose us over competitors. This wasn't just the right thing to do—it resulted in sales!

However, the number of people needing accommodations are relatively few, and implementation requires ongoing work. The harsh truth is that there are probably five other website projects that would drive more sales than adding accessibility. For nearly everyone who has a website, while "accessibility" might be priority number eight, it's not number one.

"For nearly everyone" suggests a solution. For products in purgatory, the key question isn't whether customers see value, but rather *which* customers are ready to buy *now*. They're only ready when it's a top three priority, whether driven by emergencies, strategic decisions, or company culture (as in our case at WP Engine).

So: When does "accessibility" become a major blocker for a company's revenue? Many government contracts require accessibly from their vendors, so if a company were to create a strategic mandate to expand (p. 793) their sales into the government sector, "accessibility" suddenly becomes a top-three priority, because it is required to execute that strategy. Or, companies being sued over accessibility, those selling to audiences with accessibility requirements, or those integrating with major platforms like Walmart or Amazon that mandate accessibility.

In general, you solve Purgatory by further narrowing your target market with a condition of "...and needs it right now." You then target all marketing, sales, and new product features to that ideal customer profile (p. 317).

These guiding questions will help you locate pockets of urgency:

For whom is this critical to their strategy, especially if it's directly tied to revenue?
Examples: Regulatory compliance, industry certifications, public declarations of strategy, entering a new market, buying another company, dealing with a competitive shift in their market, preparing for or just having closed new fundraising (especially IPO), modernizing legacy systems, customers publicly demanding new requirements

What kind of emergency would force a company to buy something?
Examples: Getting sued, executive turnover, a PR debacle, a negative analyst report, a competitor with a loud, large announcement, a significant employee leaving, a public security breach, regulatory investigation, market-share decline, supply-chain disruption, manufacturing defect, scaling issues

What competitive pressures create urgency?
Examples: new competitors entering their market, competitors gaining market share, competitors announcing major new features or initiatives, trade publications emphasizing things they lack, technology disruptions in their market, geographic and economic shifts

What financial pressures create urgency?
Examples: Budgets closing at a specific time of year, needing to "use up" budget, new fundraising creating new expectations together with new budget, new teams with new expectations, a change in profitability, a public change in their financial goals

How would you find customers in this condition?
Examples: Official announcements, CEOs and founders on podcasts, consistent customer complaints in reviews and social media, a shift in messaging, purchasing another company, changes in employment openings,[501] *trends in employee comments on Glassdoor, industry analyst reports, trends in trade publications, venture capital allocations, regulatory change notifications, recent leadership changes, adding new departments or teams, life events like a birth, death, marriage, divorce, changed jobs, changed homes*

There's no such thing as a product that is always valuable or always risky or always a top priority. There are market segments where those things are true, and segments where they are not.

The answer might be bad news: What if there isn't a target segment, or it's too small, or it's too hard to find them, or they have a disqualification like no budget or being locked into another vendor. Well then, it might not be workable. But this is the critical question to escape Purgatory.

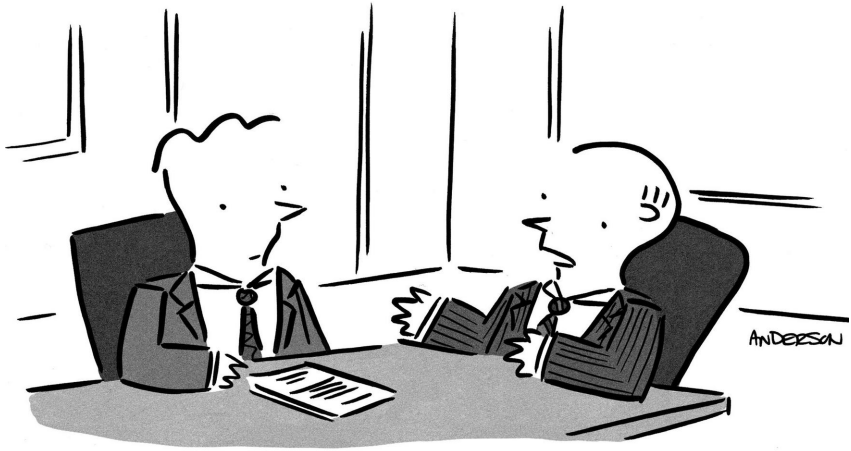
Even if you think the universe of "perfect customers" is small, take heart: For every customers who is in the center of that bullseye, there are 100x more who will find it useful enough to buy anyway. The

reason for this is not obvious; it is explained in this article (p. 317). It is scary to focus your laser on such a small surface area, but it works.

In any case you have no choice, if you want to exit Purgatory.

Chapter 30:
What makes a strategy great

SIMPLE · CANDID · DECISIVE · LEVERAGED
ASYMMETRIC · FUTURISTIC · BAD STRATEGY
MAKING STRATEGY · FURTHER READING



"I'm looking for a strategy to leverage our core competencies with big data across multiple synergized paradigms. Or something that rhymes. Either way."

credit 502

Strategy is: *How we will win.*

You can debate the form a strategy should take, whether a four-sentence "master plan"[503] or a few dozen bullets[504] or a six-pager[505] or an eighty-pager or a template like Salesforce's V2MOM[506] or a Notion template you found on the Internet or answering three questions from a Twitter pundit[507] or some sort of "Canvas."[508] Regardless, its job is to communicate "how we will win."

There are a lot of documents titled "Strategy," but very little *great* strategy. The formula for great strategy isn't decided by the format of the output document.

> *"Greatness needs luck, but it's never by accident."*
>
> *—Unknown*

For strategy to correctly determine and communicate "how we will win," it must tackle the reality of complex systems, it must parry the follies while exploiting the strengths of human nature, and it must justify and spell out the decisions which lead to the desired outcome.

Great strategies accomplish this with the following characteristics:

- **Simple:** Reshapes complexity to be manageable and actionable.
- **Candid:** Dares to spotlight the most difficult truths.
- **Decisive:** Asserts clear decisions *and* accepts their consequences.
- **Leveraged:** Magnifies strengths into durable competitive advantage.
- **Asymmetric:** Defeats uncertainty with higher upside than downside.
- **Futuristic:** Solves for the long-term.

Without these qualities, the so-called "strategy" is at best a plan; at worst, it's wishful thinking masquerading as "vision."

SIMPLE

> *"Simple can be harder than complex: You have to work hard to get your thinking clean to make it simple. But it's worth it in the end because once you get there, you can move mountains."*
>
> —*Steve Jobs*

The world is complex, and therefore difficult to reason about. Strategy intentionally omits detail in exchange for clarity. Armed with a simple narrative, mere mortals can achieve understanding, and make every-day decisions that remain aligned to that narrative.

No one reads anything. You're in the Top 1% just for reading this sentence. No one remembers anything. Certainly not an 86-slide PowerPoint.* So the strategy has to be simple—even *simplistic*—to have a chance at being read or remembered.

* Long *supporting* documents are useful, because they explain and justify complex topics, inspiring conviction even in people who don't read it but are impressed by its size. Even the famously pithy Tesla "Master Plan" strategy[509] consisting of just 39 words was preceded by 1200 words of justification.

People are mired in their day to day work; only when the strategy is simple, does it have a chance of being incorporated. Only when the context is over-simplified, having made scrutable the complexity of the real world, can we easily explain "why," and retell those stories in our weekly meetings and prioritization sessions. Those are the places where strategy lives and breathes, where teams can move quickly, independently, fulfilling their individual missions with a minimum of coordination, yet all supporting a common theory for how we will win, together.

> *"The bigger the project, the more we need a reminder of the goal, and simple things to execute today."*
>
> *–Warren Buffett, shareholder letter, 1982*

If the strategy is simple, it might appear obvious. If it is obvious, it might appear uninsightful. Humans assume that complex puzzles require complex answers, when in fact often the simplest answer is the best answer.* "Obviousness" is a sign of a strategy that not only is easy to communicate and execute, but also believed.

Its job isn't to be non-obvious, but rather to cleanly specify what is most important. It might be obvious to do X, but Y and Z are also "obvious," so by selecting X and not Y nor Z, you have created focus, and specified "how to win."

Simplicity at its worst becomes *reductive*—overlooking complexity rather than tackling it, resulting in conclusions that, while admittedly simple and clear, are wrong. We cannot pretend that complex prob-

* See, for example, the Midwit Fallacy,[510] or how small sets of equations explain a wide range of physical phenomena.

lems can always be waved away by simple statements. The process of strategy *creation* must indeed tackle the complexity of the world, but the *output* of that process is a document that frees everyone else from having to re-solve every puzzle. Like a street map, we omit detail in exchange for clarity of the most important context and routes. Simplicity that ignores reality is reductive, but simplicity that arises from an exceptional summarization of having already processed the messy, complex world, is elegant.

Characterizing the world in a few, simple, clear assertions, and solving our challenges with a few, clear directives of what we must do, is required for a strategy to be "how we will win."

Example: The most famous sports coaches seem to credit their success to a few simple principles.[511] *They must be simple for a team to remember them, especially when they're tired and battered near the end of a game, as in this description of the mentor of one of America's most famous football coaches:*

> "Blaik's signature talent was using all this data to create something clean and simple. He had what Lombardi called "the great knack" for knowing what offensive plan to use against what defense, and the "discarding the immaterial and going with the strength." All the detailed preparations resulted not in a mass of confusing statistics and plans, but in the opposite, paring away the extraneous, reducing and refining until all that was left was what was needed for that game against that team. It's a lesson Lombardi never forgot.
> —David Maraniss, *When Pride Still Mattered: A Life of Vince Lombardi*, 1999

CANDID

> *"See, and don't be afraid to see what you see."*
>
> *–Ronald Reagan*

Strategy must lay bare the most frightening, embarrassing realities. It must face the truth (p. 657) that we all avoid facing during the struggle of daily work.

If the strategy doesn't expose—and then solve—the most brutal facts, the strategy is wrong.

In particular, the strategy must diagnose the primary challenges facing the business, even if the facts are so scary that it seems hopeless. Existential threats are the most important to articulate.

Too often a strategy claims "threats" that are lazy, generic, non-actionable pseudo-concerns that nearly every company could claim, like "Google could copy us" or "A new startup could invent a great product and get huge funding" or "A massive security breach could cause half our customers to leave." Real threats are either happening now or are at least 70% likely to happen (p. 997). Real threats are specific, ideally backed by data that proves they are happening, e.g. your market is shrinking; a competitor has accelerating market share; cancellation rates prove that even paying customers don't value the product. Real threats are written in the present tense because they are happening, not hypothetically with "could" or "might."

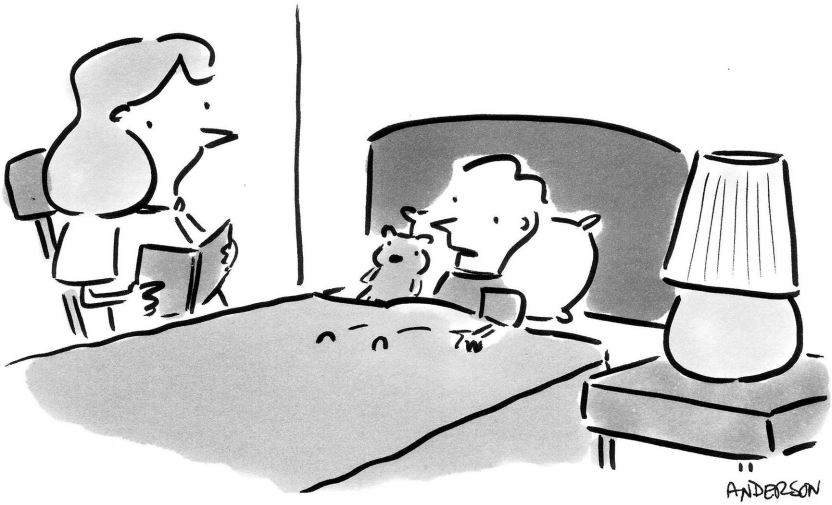
When the puzzles are clear, everyone can help solve them. When the puzzles remain hidden, what's the chance that they'll be solved by accident?

> *"Average players want to be left alone. Good players want to be coached. Great players want the coach to tell them the truth."*
>
> *–Nick Saban*[512] *(won more US national college football titles than any other coach)*

If you're worried someone might leave the company when they hear how scary it is, maybe they *should* leave the company. These *are* the challenges we're facing together; if they're not willing to solve them, they need to make space for someone who will relish the challenge. Not just for the company's sake but for their own happiness and fulfillment (p. 399).

Facing the truth, being specific about current reality and about what needs to be done, is required for a strategy to be "how we will win."

Example: In a story retold here (p. 657), originally from Jim Collins's book Good to Great, A&P and Kroger were successful grocery stores in the 1960s who both possessed the data showing that their business models were becoming obsolete. They each discovered the new, correct business model through real-world experimentation, but only Kroger was willing to face that truth and change their strategy. Despite being half the size of A&P, Kroger increased its value 100-fold, while A&P shrank, then went bankrupt.



credit[513]

"Let's say honesty isn't going to work. What's the *second* best policy?"

DECISIVE

A strategy asserts a set of justified, self-consistent decisions, such as:

- Which subsets of the market to target
  ...and conversely which we'll ignore, even if some of those sign up as customers anyway, and ask for things we're not going to do, and then cancel in anger
- Which customer personas are most important to delight
  ...and conversely which will dislike our product, causing us no dismay, whose feature-requests we will quickly close as "won't do" rather than wring our hands at all the features we still need to build

- How to position against the competition
  ...and conversely where the competition will be stronger, unlike those fake-news marketing charts comparing our product with the competition, where only our product scores 100% along every dimension
- What we value (e.g. quality, service, speed, design, compatibility, lock-in)
  ...and conversely what we will give up, e.g. releasing features faster but of lower quality, or being infinitely extensible versus top-to-bottom thoughtful design
- What we must *(and must not)* build, to pay off that positioning and win those customers in that market.

> *"You can do anything, but not everything."*
> *–David Allen*

There are both positive and negative second-order consequences of any complex decision. If these aren't identified—in particular, if the problematic consequences aren't embraced within the strategy—then it's not a clear decision. When those consequences inevitably arise, the team must be able to say "we expected that" rather than "we have to address that" or even "this is a signal that the strategy is wrong."

One decision in every strategy is what market and customer-segment the company will target. Customers outside that target segment will inevitably buy anyway. Then they'll complain about missing features, awkward UX workflows, missing integrations, high prices, and more. Then you'll log those complaints into the bug-tracker and product-management Miro boards and start prioritizing, whereas in fact these ideas should be ignored so that the team can focus on winning the target customers. This is difficult to remember when the non-target customers complain on Twitter, supply low NPS scores, drag

down your average star-rating on review sites, and cancel at high rates. Only with a strategy that has clearly articulated not only the decisions, but also the negative consequences of those decisions, can everyone stay true to those decisions even when emotions are running high and paying customers are canceling.

A common tactic for avoiding making a decision is to use non-specific language. "We will leverage synergies to create unique solutions" is, in fact, a good thing to do, but it doesn't specify which synergies to leverage, what is unique (p. 891) about it, or what the unique solution is. Fluffy language (p. 627) is a hallmark of indecisiveness, and therefore of bad strategy. Being specific is good marketing (p. 1439), anyway.

> *"Strategy is a set of interrelated and powerful choices that positions the organization to win."*
>
> *–Roger Martin*[514]

The opposite of the decision must also be a rational choice, made by other successful organizations. For example, deciding to be open source is strategic, because plenty of companies are successful with a closed-source strategy. However, deciding to be "customer-first" is not a serious decision, because successful companies don't use a "customer-last" strategy. This "Opposite Test (p. 1589)" is useful both to form proper strategic decisions, and to form great positioning statements for marketing.

Decisions and consequences must at minimum be self-consistent. If you've decided to have a low price, you can't also have white-glove service. If you've decided that everything requires high-quality design, you can't also release features faster than the competition. Or, perhaps

you can creatively build a solution that does say "yes to both" of those things, but only by also accepting additional constraints that resolve the conflict (p. 589).

Better than "self-consistent" is "mutually-reinforcing." This means that one decision makes another more powerful, or easier, or less expensive, and vice versa, so that adhering to both makes you far stronger than having only one. For example, deciding to have only a few features, and also amazing design. Normally customers might not put up with less functionality, but if the design experience is exceptional, they might be happy with something that "does only a few things, but so delightfully!" And vice versa: It's easier to execute on great design when you don't have to tackle a complex product with tons of use-cases and personas and functionality. At the end of this article (p. 543), you'll find a complete, powerful example of mutually-reinforcing decisions creating a durable (60-years!) competitive advantage.

> *"The difference between successful people and really successful people is that really successful people say no to almost everything."*
>
> *–Warren Buffett*

You know the decisions are strong when they—and their consequences—cause you to say "no" to most things (p. 621), including things which otherwise sound reasonable. An example of a strategy that makes a decision but also accepts negative consequences can be found at the end of this article about Moats (p. 761); an example of how to make sales pitches while accepting negative consequences can be found in this article;[515] several examples of how YouTube used a

credit[517]

"I haven't yet decided if I'm staunchly in favor, or staunchly opposed. But rest assured, however I vote, it will be staunchly."

single, strong decision with multiple major downstream consequences are detailed in this article.[516]

When a strategy has articulated clear decisions, including the major consequences, especially accepting the negative consequences, causing us to say "no" to pretty-good ideas in order to make room for the very-best ideas (p. 603), and to ignore the wishes of paying customers if they're not the target paying customers, it is truly prescribing "how we will win."

Example: Craig's List[518] *is classified ads on the Internet. Their strategy includes crisp decisions like valuing consistent user experience more than great design (the website looks like it's from 1995, because it is), and user-accessibility over monetization ("Craigslist president Jim Buckmaster has stated that creating a superior user experience is more important to the company than making money [source*[519]*]"). The strategy is clear and differentiated, which should be seen as a strength, but it has bothered pundits for twenty-five years, who have therefore predicted its demise: "The main*

obstacle to sustainability and growth at Craigslist is likely the company's and its founder's strong principles valuing customer-offering over monetization, trusting consistency over innovation. Although admirable in many ways, the issue is that without innovation, the company's customer-offering will soon no longer be strong enough to stay relevant, undermining its very well-intended purpose."–HBS.[520] *But crisp decisions make for a great strategy, even if controversial. Despite that well-reasoned death-sentence from HBS, despite others pointing out that dozens of successful companies have*[521] *made sections of Craig's List theoretically obsolete, despite data that predicts Craig's List "demise",*[522] *despite protestations*[523] *that Craig's List could be a "gold mine of revenue, if only it would abandon its communist manifesto," (i.e. if only it would abandon its strategy), it continues to be one of the most-visited sites in America,*[524] *with revenues of $1,000,000,000 in 2020 with a staff of just fifty people, even after twenty-five years of Internet evolution.*

LEVERAGED

> *"Be yourself. Everyone else is taken."*
> *–Oscar Wilde*

If Sandra is a great singer, and Peter is a great pianist, then a good idea for a duet would be for Sandra to sing and Peter to play piano. A bad idea would be the other way around. Great strategy leverages the strengths and assets of the organization; a bad strategy asks the orga-

nization to win even while acting unnaturally, often under the cover of "overcoming weaknesses."

"Leverage" means generating a large effect from a relatively small effort, where time and dollars are far more effective than one might expect, because we are riding tailwinds of natural abilities or hard-won assets, rather than fighting a battle against so-called "weaknesses." You know you're leveraging strengths when you see other people shake their heads in amazement at how much you accomplished in so little time.

It's good to leverage strengths. Much literature on strategy dwells on how to create moats (p. 761)—permanent competitive advantages—but so many organizations still aren't leveraging the straightforward, undifferentiated strengths that they possess. They expend most of their energy shoring up "weaknesses," which despite their efforts will at best become "less weak," but never become a strength. Whereas applying that same energy in leveraging their strengths will have a large positive effect. It's even just more *fun* to play to your strengths instead of wallowing in weaknesses.

It is of course better when the strength is *differentiated* from the competition. This is especially obvious in a mature market where everyone is saying the same things on their home page, pricing the same way, and different only in tertiary characteristics. Winning in your own way can defeat "better" competitors (p. 891).

It's even better when that differentiation is *durable* over time. It has never been more difficult to establish a permanent advantage, when all software can be reproduced, all business models can be replicated, and the entire world is both your market and your competitors, which makes it all the more important to decide what one or two moats you will build. The strategy is the place to name those moats.

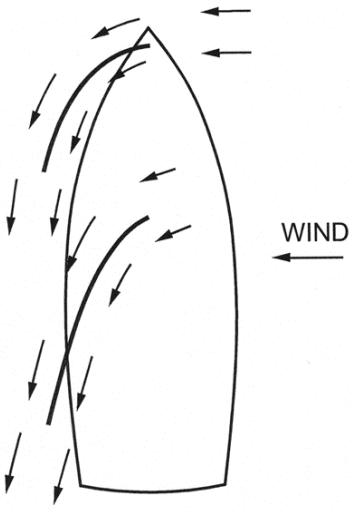
This companion article on "leverage" **(p. 543)** provides expanded examples and ideas for all of the above.

Tailoring the decisions for the strengths of this organization, avoiding (rather than reversing) weaknesses, even better when the strength is differentiated, identifying and investing in durable differentiation,

so that moats are constructed in the long run, is required for a strategy to be "how we will win."

Example: Zappos started in 1999—early Internet, no social media, few people with broadband, few people comfortable buying anything online, much less buying shoes, which for 100 years was bought only after first trying them on in person. Zappos started with a differentiated insight (the companion article explains how "insights" can be leverage): That people would actually buy shoes online, under the right circumstances. They built a moat out of a combination of surprising decisions (also explained in the companion article): (a) postage-paid returns even 364 days after purchase, (b) legendary support who did things like buying pizzas for customers, (c) a corporate culture of employee empowerment; it takes that combination of strong, expensive decisions to create the conditions by which people would dare to buy online, thus unlocking an entirely new retail channel. Only by having all three of those components did the moat work; if you don't empower employees, they can't order pizza on the spur of the moment; if your return policy isn't ridiculously generous, people will be too scared to order shoes sight-unseen. Because their vision and decisions were so incredible (literally "not credible"), competitors laughed at them while they enjoyed a constant stream of positive press. The result was the most successful online shoe store for a decade, reaching annual revenues of $1B before its tenth birthday.

ASYMMETRIC



A sailboat always moves forward.

This remarkable fact is due to its asymmetrical shape: it is pointy at the front and flat in the back. This creates resistance to moving backwards, but a natural ease in moving forward. Even when the water is randomly undulating, "backward" forces are muted, while "forward" forces are allowed, so the boat glides forward. Asymmetries can amplify positive effects while muting negative ones, resulting in a net-positive force even under conditions of random inputs.

Great strategies prescribe activities that always move the company forward, despite the inevitable bad luck and setbacks. To do that, the activities must exhibit asymmetry, where the upside vastly exceeds the cost, so that even if you took 2x longer to achieve 50% of what you expected (p. 171), you still win.

> *"If riskier investments could be counted on to produce higher returns, they wouldn't be riskier."*
>
> *–Howard Marks*

This is the mechanism behind Venture Capital portfolios, which are investments in a slate of early-stage startups. The worst-case outcome for each bet is that they lose 100% of their investment, but in the best case they can gain 10,000%. A few large successes more than make up for the many failures (p. 1261), so the portfolio in total comes out positive. Investors call these "asymmetric bets." Economists call this "convexity."

Strategy must create a portfolio of bets having this "VC-like" asymmetric quality, whether for a small startup trying to find product/market fit or a mature company entering new markets. A sign of a bad strategy is when success requires everything to go right. With a set of asymmetric bets, the successes render the failures moot, and so the unpredictable waves crashing into the boat still result in forward motion.

This companion article details strategies that work well in an unpredictable world (p. 193). One of the few strengths of a new startup is that it can learn faster, react faster, deliver faster (p. 295); startups are favored when the waters are uncertain.

One form of asymmetric bet is entering a large and growing market. Besides the obvious benefits* there is the asymmetry of optionality: There are many niches to exploit, many possible ways for a product to deliver value, many marketing and sales channels, and there's more of all of it every year. Because there are many options to try, there are many ways to succeed; if your first few ideas don't work, the next one might. By having lots of ways to succeed (p. 1277), you are more likely to find one. Waves on the boat.

* Potential customers are already spending money, which means budgets are preallocated and pricing structures are well-understood, the press is already talking about it, marketing channels already exist, and the "pie" is growing, so even a small slice of the pie automatically grows.

> *"The most powerful force in the universe is compound interest."*
>
> *–Albert Einstein*

Another kind of asymmetry is a process that compounds, meaning that the more of it there is, the faster it grows. Things with this characteristic naturally grow larger than anything that grows in a more linear fashion, even if they start out small. Examples are customer retention, customer upgrades, and employee retention. Another example is a growth-vector that is proportional to the size of the current customer base, such as word of mouth referrals and viral products (e.g. once you join a social platform or collaborative online tool, you tend to get other people you know to join as well). Most things that grow non-linearly don't grow exponentially (p. 115)—that's normal, and still a great strategy. A common example is a reseller channel, because each reseller could bring you a number of clients over time, so when the number of resellers grows by `N`, the number of potential customers they bring grows by `N × M`. For example, Intuit grew Quickbooks by providing software to CPAs that made their life easy, but only if their clients used Quickbooks. The CPAs started requiring their clients to buy Quickbooks, and so a smaller number of CPAs created a much larger number of Quickbooks sales.

In the negative, companies also face asymmetric threats. With ten competitors, only a few need to become break-out winners to pose a significant challenge. Like the famous statement by the IRA after a bombing[525] failed to kill British prime minister Margaret Thatcher: "Today we were unlucky, but remember we only have to be lucky once. You will have to be lucky always." A strategy that places a number of asymmetric bets doesn't need to be lucky always.

Every plan will face challenges, both foreseeable and bad luck. If everything has to go right for the plan to succeed, it won't succeed.

Whether a single investment has asymmetric upside, or a portfolio of bets collectively has large upside, exploiting asymmetries maximizes the chance that the strategy will succeed despite the inevitable travails (p. 429) and uncontrollable luck (p. 1035), and thus is vital to "how we will win."

Example: Amazon is the cliché example, but only because it's apropos, even ignoring AWS. Originally a book-seller only, they have never stopped taking bets, whether on adjacent markets (e.g. selling electronics), reselling internal systems (e.g. Amazon warehouses, logistics, robotics, fulfillment, were all broken apart and sold as separate products), or entirely new kinds of product (smart phones, cloud computing). More impressive is when the new bets purposefully disrupted previous businesses; according to early employee Andy Johns (article,[526] *podcast*[527]*), Bezos was so adamant that the Kindle be successful, despite the fact that it would hurt their physical book business, that he assigned the Kindle project to the executive who was currently over the book business, saying "Effective tomorrow, your job is to kill your old business with a Kindle." The key thing is:* ***many of these bets failed, and failure was expected.*** *Failure isn't desirable, but each bet had out-sized potential upside and budget-able downside. Even Kindle was unprofitable for years*[528] *and the Amazon Fire Phone was an abject failure.*[529] *Bezos explains in his own words that their strategy is exactly the theory of a portfolio of asymmetric bets with expected failures:*

> "If you're going to take bold bets, they're going to be experiments. And if they're experiments you don't know ahead of time if they're going to work. Experiments are by their very nature prone to failure. But a few big successes compensate for dozens and dozens of things that didn't work."
>
> "I've made billions of dollars of failures at Amazon.com. Literally. You have to be super clear about what kind of company you're trying to build. ... We said we were going to take big bets. We said we were going to fail."
>
> –Jeff Bezos, The Guardian,[530] 2014

FUTURISTIC

> *"The best time to plant a tree is one hundred years ago. The second-best time to plant a tree is today."*
>
> *–A proverb claimed by many peoples, across thousands of years*

"Being agile"* is a great way to climb the proverbial mountain-shrouded-in-fog. Some paths are the right ones, but backtracking is inevitable; it's a sign of puzzling-out, not a sign of failure. The job of strategy is to identify which mountain we're trying to climb in the first place—the puzzle we're solving for, the opportunity we're exploiting. If an "agile, self-managed" team climbs the wrong mountain, it was all a waste.

Strategy looks further into the future than anything else at the company. Therefore, it has the responsibility to take the long view. Which is especially difficult, as the future is unpredictable (p. 193), and data tells you about the past, but rarely about the future.[531]

If you can solve a problem in a month, you probably should, but that also means it's not a *strategic* problem. Anything that can be

* Meaning: Don't pretend you can predict the future, assume the quantity of "things we don't know" is larger than the quantity "things we do know," iterate quickly on hypotheses (p. 239) that you proactively attempt to disprove, and adjust in the presence of new information. The team that learns the fastest, wins. The team that spends three months trying predict the future, is now three months behind, and the future still won't unfold as they predicted.

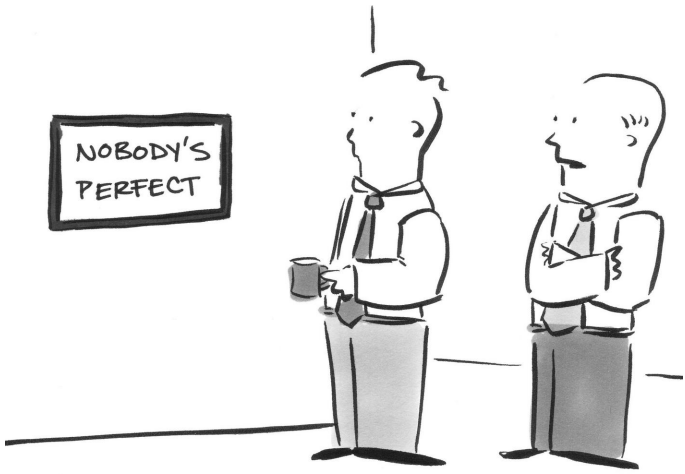
built in three months, isn't the way you will have constructed a moat that will take competitors years to overcome. Anything that specifies features or timelines is a roadmap, not a strategy; a strategy specifies market and business outcomes and the primary decisions and secondary consequences. Anything that specifies teams or roles or hiring or processes is an operational plan, not a strategy; a strategy details the outcomes and activities that require the entire company to accomplish together, not what one team needs to accomplish alone.

People naturally get distracted by the immediate, the urgent, the tactical, the "low-hanging fruit." Strategy is the place where we select the high-hanging fruit, the "important, but not urgent" quadrant of the Eisenhower Matrix.[532] Then executed as the Rocks in your "Rocks, Pebbles, Sand" prioritization framework (p. 221).

Part of explaining "how to win" is defining what the "winning" state looks like. Often called the "vision statement," it describes what the world will look like when we're successful:

Oxfam	A just world without poverty
Habitat for Humanity	A world where everyone has a decent place to live
Stripe	Increase the GDP of the Internet
Microsoft (years ago)	A computer on every desk and in every home
Tesla (original)	Create the most compelling car company of the 21st century by driving the world's transition to electric vehicles
WP Engine	Power the freedom to create online
YouTube	Give everyone a voice and show them the world
GoDaddy	Radically shift the global economy toward independent entrepreneurial ventures

The Vision statement is often the first sentence the strategy document, but was the last thing crafted by the authors of that document.



credit[533]

"You're right. It's not much of a mission statement."

Only once you fully understand the challenges you face, the main, coherent courses of actions to undertake, and the results you want, can you summarize a clear vision of what the future will be.

Because words like "mission" and "vision" (p. 1169)have indefinite meaning and are often misused, I prefer alternate words and intentions, as described in the article just linked.

Strategy is where we specify the most critical long-term challenges facing the company, and the rocks (p. 221) that are the most important things (p. 603), not to win the battles today, but to determine how we will have won the war three years from now, how we will achieve our vision, how we will win.

Example: After Google purchased YouTube, they set a far-future goal of attaining a billion daily views. This filtered down to everything from demands on technical architecture, to strategic debates such as whether that enormous quantity of views could be achieved by user-generated content alone or whether they needed to license libraries of existing TV shows and movies. At one point they realized that the global Internet network infra-

structure did not have enough capacity to stream a billion views per day, so Google invested billions of dollars in fiber and data centers to prevent that from becoming a bottleneck—billion of dollars they would not have spent without a vision of the future that mandated that investment.

BAD STRATEGY

Tell-tale signs of a strategy that lacks these qualities:

Not simple
Pages of detail.* Slides with more than 20 words. A litany of numbers without a narrative explaining what insights they create. No diagrams "painting the picture," or diagrams with 20 boxes. Too many points for someone to recall from memory.[534] Important concepts that aren't summarized by a short phrase that people can use as a daily short-hand.

Not candid
Nothing where the future of the company hangs in the balance. Nothing that makes the reader say, "Oh wow, dang, what are we going to do about *that!?*" Nothing scary that demands action. No serious consequence if the directives aren't followed. A reader who finishes the document and thinks, "We're still ignoring the elephant in the room."

Not decisive
No clear decisions that would cause us to "easily say 'no'" to many otherwise excellent, reasonable ideas. Not obvious what we're *not* doing. Non-specific target market (e.g. "for everyone"), target customer (e.g. "any [title]"), target jobs-to-be-done. Directives and headings using the word "and" to expand scope rather than limiting it. No negative-but-accepted

* Detail is great, if it is attached as optional reading, expanding on a summary.

consequences of the decisions. Decisions that conflict with each other. Decisions that don't reinforce each other.

Not leveraged
Strategy would apply equally well to a competitor, or even to a business in another industry. Strategy doesn't call out the special strengths and durable assets of the organization, or doesn't explain how to apply them to win, especially how it will position against the competition. No obvious moats being constructed. No network of interlocking decisions that together makes the company special. Demanding that the organization overcome more than one or two major deficiencies.

Not asymmetric
Potential upsides aren't at least 10x larger than costs. Linear cost/reward, or risk/reward. Not creating optionality in how each aspect can go right, leading to one thing that absolutely must work. A course of action that requires multiple different, difficult things to simultaneously go right, otherwise the whole strategy fails.

Not futuristic
Doesn't describe a specific future destination of the company or product. A "vision" that describes what the company already does and already is, rather than how things will be different once we successfully execute our strategy. Doesn't specify which moats are being created, and how. Specifies teams, products, timelines, or features. Deals with temporary challenges that can be solved in a quarter rather than long-term challenges that will take years to fully overcome. Describes how to win this year instead of in three years. Relies primarily on data to predict the future.

Not strategy (bonus)
Generic statements that would apply equally to nearly any company, even in a different field (e.g. grow faster, lower attrition, hire the best talent, delight customers, beat the competition). Aspirations about what we wish would happen, without specifying how it will happen. Financial goals rather than how to win competitive markets. Plans that are in someone's head instead of written down and shared. Written documents that aren't referenced when creating plans.

A sailboat always moves forward, but if you don't decide on a specific destination, it will end up *somewhere*, but probably not where you wanted to be.

Time to decide how you will win.

POSTSCRIPT: HOW DO I CONSTRUCT A STRATEGY?

Stay tuned for future articles that lay out a process for constructing a great strategy. Partial answers appear in The roadmap for Product/Market Fit (p. 9) and Excuse me, is there a problem? (p. 71).

But don't wait. A mediocre strategy is still better than no strategy.

FURTHER READING

- Article: WTF is Strategy?[535] by Vince Law: Defining the nested concepts of vision, mission, strategy, roadmap, and execution, thereby contextualizing the role of "strategy."
- Book: *Good Strategy, Bad Strategy*[536] by Richard Rumelt: Doesn't explain how to construct a strategy, but terrific observations about what good and bad strategy looks like. (Summary[537] by Jeff Zych)
- Book: *Blue Ocean Strategy*[538] by W. Chan Kim & Renée Mauborgne: Explains how to construct a strategy that is not only "different from the competition" (i.e. competes better in so-called

"red-ocean" force competitive markets) but competes in a wholly different way, creating a new value-proposition, often at higher profit, in so-called "blue oceans" where you've defined yourself such that there isn't any direct competition.

- Article: Gibson Biddle's multi-part series[539] on how to build a product strategy that converts "how to win" into "what to do"
- Article: Taylor Pearson on optionality,[540] and how to apply the Nassim Taleb quote in practice, even personally.
- Article: Excuse me, is there a problem? (p. 71) on selecting the right "problem to solve" and "for whom" to increase the probability of success.
- Article: Using the Needs Stack for competitive strategy (p. 259) on thinking about customer value, markets, competitors, and disruption.
- Article: When customers are "willing" to pay (p. 275) on how to create strategies that capture and keep customers as your allies, who help you grow and find new customers, rather than forcing them to stay through coercion.
- Journal article: The Design School: Reconsideration of the Basic Premises of Strategic Management[541] by Henry Mintzberg, *Strategic Management Journal* (1990), with a summary of traditional strategy construction, a critique of its drawbacks, and conditions when it is appropriate.

Chapter 31:
Pricing determines your business model

"We're not changing anything, we just wanted to charge more."

credit[542]

It's often said that you shouldn't talk about price during customer development interviews (p. 239). The usual justification is that your goal is to uncover the details of your potential customers' lives and pain-points, whereas mentioning price diverts the discussion to budgets and operational things far removed from the customers' day-to-day life.

But I disagree. Price is as important as any other feature to determine product/market "fit."

How many times have you heard someone agree that "it would be great if someone did X," but when you show them a demo of X, but it costs $700, and they don't buy? Or seen a review of an iPhone app hung up on pricing trivialities: "It would be pretty good at $0.99, but it's not worth $1.99." How many times have you seen someone struggle with an inferior product because they cannot afford the better one? Or struggle with the freemium version because they refuse to pay anything at all, even though they like and use the product? Or struggle with an inferior, expensive product that was purchased based on the salesmanship—"expensive must mean it's better"—instead of craftsmanship?

Price is inextricably linked to brand, product, and purchasing decisions—by whom, why, how, and when. Price is not an exercise in maximizing some micro-economic supply/demand curve, slapped *post facto* onto the product. Rather, it fundamentally determines the nature of the product and the structure of the business that produces it.

Consider the consequences of each order-of-magnitude of pricing:

$0/mo means your goal is to maximize growth (trust and usage) instead of revenue. Your product is designed with natural tripwires to trigger other pricing (Freemium model (p. 1385)), or not (business model left as an exercise to your future self). Typically requires venture funding because you have no income, and if you're successful you'll need lots of people and tech to run the business. Even super-efficient outliers like WhatsApp (sold for $19B with 55 employees) and Instagram (sold for $1B with 13 employees) each raised tens of millions

of dollars in venture funding. This is often B2C because the value is in quantity of customers, and there are 100x more consumers than businesses.

$1/mo means you can't afford customer service and it must incrementally free to run the technology behind it, both of which have implications for the sort of product you have to build (e.g. simple enough to be self-service). Marketing and sales spend is nil (at least initially), so there has to be a reason it spreads by word of mouth, ideally virally (p. 115) as a natural result of using the product itself, or through vocally passionate early users, perhaps galvanized around a mission.

$10/mo means people see you as a cheap version of something else, but still expect a phone number. (Think: GoDaddy). Bootstrapped businesses can make this work (e.g. most shared hosting companies), but they only make interesting money at large scale (by definition, because it takes over 8000 customers to make only $1m/yr in revenue),

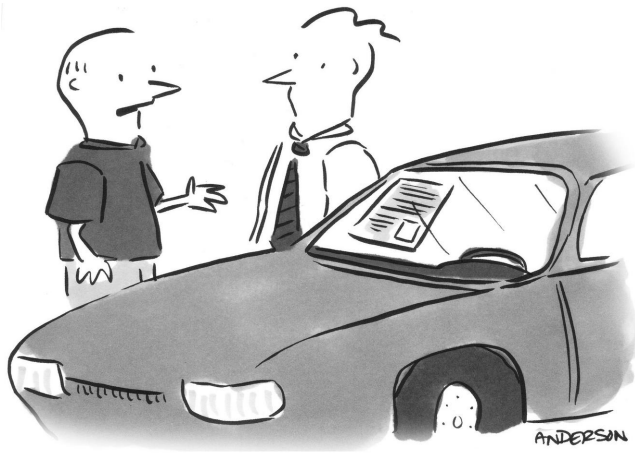
"And for the connoisseur on a budget we have a lovely Tuesday vintage at a buck fifty."

credit[543]

which takes a long time to grow. So you can get to $10s of millions but it will take 10 years. (Again, like shared hosting companies.) This is a hard slog. If you want to scale faster you'll need venture funding, both because of the anemic revenue, and because otherwise you can't afford to advertise. Often bootstrapped companies of this type boast about having no marketing or sales departments, but the truth is they can't afford it, and those companies typically grow slowly, often eclipsed by companies who can afford to grow 10x faster. In a huge market this is probably still OK because there's enough customers for everyone to thrive in different ways. On the good side for this business model, often people will simply forget they have the $10/mo service even if they don't use it, resulting in "free revenue" which at scale can be surprisingly substantial (yet again like shared hosting companies).

$100/mo means people expect to be able to call support, and if a competitor is substantially better, it's probably worth the effort to switch. Also this is almost exclusively B2B unless it's something "luxury." Big companies can buy it without much consideration, but small companies need to understand the value, so you might need sales material to convince them, and a demo. I love this price zone for bootstrapped companies, because it's low enough that you can address a broad business market, but high enough that you can get nicely profitable with a reasonable, achievable number of customers (e.g. 200-300), and you can afford to spend money acquiring them (p. 1377).

$1,000/mo means only medium to large companies can afford it. They'll have buying processes around annual budgets, approvals, ROIs,[544] demos. You will be compared to alternatives and weighed. You'll be asked to give a discount (p. 895). You need to be a part of that conversation which means a real sales force, sales materials, impressive logos, case studies, and referenceable customers. Features might need to include things that big companies need that others don't, like role-based access privileges and integration with LDAP. This can be a surprisingly difficult zone to become profitable in,[545] because the



"I like it, but I'm looking for more of a status symbol. Any way you can double the price?"

credit[546]

sales and marketing motions and engineering costs are the same as for much larger sales, but without the attendant revenue.

$10,000/mo means larger companies only. It's unlikely the product is sufficient out-of-the-box, so you might need in-house professional services or to partner with consulting firms for implementation—constructing the "Whole Product" in the language of Geoffrey Moore.[547] If something goes wrong they'll cancel and not be willing to pay out the rest (no matter what your contract says; what are you going to do, sue your customers?). It's possible to achieve this price-point with mid-sized companies if it's usage-based (e.g. ZenDesk, JIRA, Box, Slack) or performance-based (e.g. marketing optimization (p. 165)), so the product itself needs to create and then demonstrate value to earn that result.

$100,000/mo means Global 2000 only, large-scale projects that require multiple departments for decision and approval, long sales

cycles (9-18 months) which requires massive cash outlay as you bide your time. Likely starts as a pilot or proof-of-concept, which hopefully you can get paid for, and which needs to be convincingly successful, perhaps battling against another pilot with another vendor. You'll need on-site visits even in this day and age of video-conferencing. Examples: WorkDay (much of revenue is consulting), IBM.

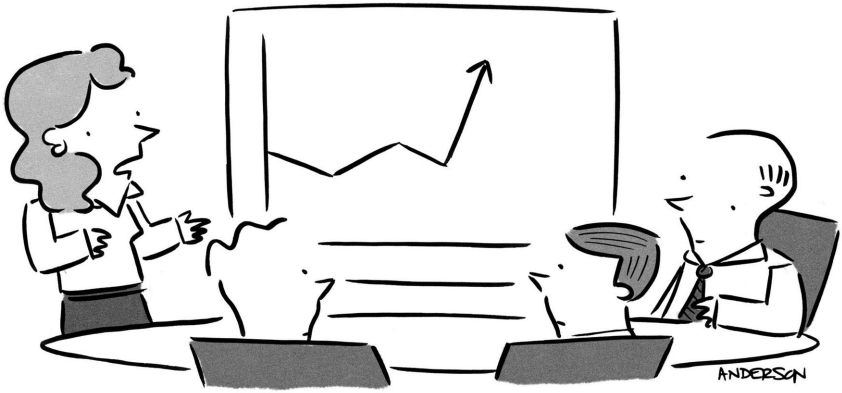
So which is better—higher or lower or in the middle? That's a better question. Here's something resembling an answer (p. 71).

The main thing is to understand that price is linked to everything, and bring it into your customer interview process (p. 239). Talk about their expectations of cost, and why, who would write the check, who would need to approve it, whether an ROI argument would be welcomed or scoffed at,[548] or what would need to be changed (if anything?) to justify doubling your price[549] (at which point, maybe you should do that an in fact double your price (p. 165)).

Price is not an afterthought, it is essential business design.

Chapter 32:
Explore vs Execute

WHAT GOT US HERE · EXPLORE VS EXECUTE
CULTURAL TRANSITION



"Things are looking good, but we need to be careful to avoid any feelings of pride. Remember what happened last time?"

THE ARROGANCE OF "WHAT GOT US HERE WILL GET US THERE"

Founders are arrogant by necessity. Most startups fail, and yet these cocky founders are absolutely sure they are the exception. "Rules don't apply to outliers like me," they scoff.

(Actually most *aren't* so certain—the honest ones are scared shitless, hoping they can figure it out (p. 429) before everyone finds out they're a fraud (p. 457) and they crack (p. 737). I find that the ones who are genuinely, deeply confident that their success is assured, are most likely to fail; they are disconnected with reality. Reality wins.)

Founder arrogance peaks once the company achieves some success, having entered the vaunted Product/Market Fit (p. 335). It was such a hard scrabble to get here, against all odds, and now it's actually *working!* Sure there was luck (p. 1035), but mostly there were many good decisions, and a great product that solves a real problem (p. 71), and decent marketing to a decent market, as evidenced by new customers showing up every day and paying for it. Incontrovertible evidence that *we were right!*

What Arrogant Founder doesn't realize is that the needs of the company dramatically change after Product/Market Fit. Everyone and everything at the company will change. Including the founder.

This doesn't make sense to Arrogant Founder, because surely the evidence points towards the opposite conclusion:

> We achieved our success *because* of the way we've run the company, the decisions we made, the trade-offs we took, the style of work we developed, the team we built, our choice in features versus design versus fixing bugs versus security versus risk. We did it by looking at what was right in front of us in-

> stead of pretending that we could build an annual plan. We did it without layers of management and without finance nor HR.
>
> Therefore, we can avoid doing the dumb things that big companies do. We can avoid becoming the kind of company we quit in order to start this one. We got successful because of certain things, and we will continue to be successful because of those things.

And thus the company descends into tumult, frustration, chaos, occasionally even irreparable disaster. Here—after Product/Market Fit but before maturity—is the most difficult part of the journey. The founder insists that "starting the company" was the hardest part, because (a) it *was* hard and (b) it was the hardest thing *so far*. But being hard doesn't make it the hardest. What's hardest, comes next.

The immediate reason things get difficult is the company is suddenly growing faster than it can manage. As just one example, there are more support requests than there is capacity to answer,* so queues grow without bound while service quality plummets.** Hiring both quickly and with quality is difficult and unlikely. Even when you succeed, it takes six months before a person you reached out to today makes it through the interview process, quits their current job with notice,

---

* At least not with the same care, quality, and personal attention that "got us here," often one of the differentiating factors that a small company leverages to beat larger competitors (p. 295).

** In January 2012 at WP Engine, when we hit Product/Market Fit as documented in a separate article (p. 9), our Twitter feedback went from "Best support team in the world!" to "Wow, what happened? They're crap now." It was 2013 before we had enough people and process, both the science of metrics and the art of culture and empathy. Then over the next decade we were known for the best service in the industry, winning many awards, with a CSAT of 98% and an NPS of 64.

takes a week off, gets trained (without training materials, while everyone else is still overloaded), and learns enough on the job to finally be productive on their own. Cross-apply that to engineering, design, product, marketing, sales, and finance.

You're overwhelmed. Digging out makes it worse, and because it happened suddenly, there's quite literally no time in which to fix it.

Beyond this immediate challenge, a dozen **facets of the business start breaking** because of scale. Not just software and infrastructure, but people and processes and workflows. Many of these facets are detailed in this article (p. 773).

So, what needs to be done? A radical shift in working style, from "Explore" to "Execute":

Explore
Figuring out what works, as quickly and flexibly as possible.

Execute
Becoming excellent at doing what works, while scaling.

You might have heard this dichotomy as "Explore vs Exploit", because that is the normal term of art used by great strategic thinkers like Roger L. Martin,[551] Clayton Christensen,[552] and James G. March.[553] You can call it "exploit" if you wish, especially when searching Google or chatting with AI. But I don't like that term, because while I'm a red-blooded capitalist like the best of them,[554] I don't like the aspect of capitalism that exploits employees, exploits customers, exploits resources, or exploits communities. "Exploitation" is when you extract value *at the expense* of another, rather than maximizing one the best things about capitalism: Creating more value than you consume, such that both parties are better off after transacting.

Anyway, "Execute" is more accurate. To meet the challenges suggested above, you don't take advantage of people or goodwill, but rather you move from a modality where you're "figuring out who we are and how we will win in our own way" into a modality where you've figured that out; now you're operationalizing those discovered

things: making repeatable, making teachable, making higher quality, automating, and building teams that themselves improve and grow. In short, becoming *excellent at execution*.

Applying the wrong modality doesn't work, in either direction. "Scaling too soon" is a common cause of startup failure (p. 379)—switching from "Explore" to "Execute" before the exploration is complete. Prior to Product/Market Fit, you don't yet know what ought to be repeated, what should be excellent, and you have to discover and jettison whatever is wrong as quickly as possible. You don't know which customer segment should be targeted with advertisements and new features. You don't know what price to charge or why someone should pick you over a competitor.

Having identified the perils of applying the wrong modality in either direction, we now need a clear, detailed definition of each mode, so the entire organization can align.

EXPLORE VS EXECUTE

Explore		**Execute**
experiment	→	standardize
fail fast	→	fail never
speed	→	reliability
creativity	→	stability
learning	→	leveraging
gathering evidence	→	data extrapolation
manual labor	→	automation
discontinuous jumps	→	continuous optimization
question assumptions	→	leverage assumptions

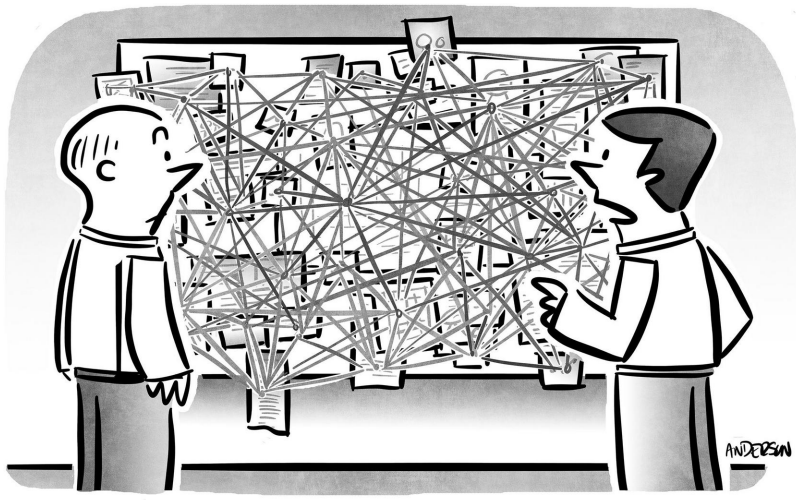
Explore		Execute
short-term tactical planning	→	long-term strategic planning
low quality buys speed	→	high quality buys safety
new	→	valuable

Before Product/Market Fit, you're figuring out what works. That means answering questions like:

- What is the problem we solve? (p. 71)
- What is our target market? (p. 1211)
- Who is the perfect customer? (p. 317)
- Why aren't all of them buying? (p. 1541)
- What makes them *want* to buy? (p. 275)
- Which few features make them buy? (p. 1271)
- What is the customer trying to accomplish? (p. 259)
- What does the customer value? (p. 165)
- What is delightful? (p. 1517)
- What distribution channel will work? (p. 1027)
- What makes us different?[555]
- What is the right price? (p. 515)
- What are our biggest risks? (p. 667)
- How do we avoid the obvious fatal mistakes? (p. 379)

The "Explore" modality is appropriate for answering these questions.

Trying, testing, failing quickly (p. 1261) when it's wrong, which it often is. Moving quickly, because we're running out of time and money. Undoing errors (p. 1045). Looking for big shifts rather than subtle changes that are too small to be noticed (p. 913), especially since we don't have enough data to actually measure those changes. Being confident about nothing;[557] questioning everything; seeking the difficult truths (p. 657).



"At first I was trying to solve a crime. Now I think I'm making an afghan."

credit[556]

Exploration does not follow a timeline nor takes a singular path, although leveraging well-researched patterns (p. 9) might increase the chance of success.

After Product/Market Fit, that list of questions have been answered.

Let that sink in. All that work, all that worry, all the uncertainty, all the probing, all the interviewing, all the experiments, all the pivots, all the doubt from yourself and others... none of that is relevant now. In fact, it's getting in the way of what needs to be done, which is to execute.

The "Execute" modality is required to answer the new challenges:

Hiring & managing, not building things only

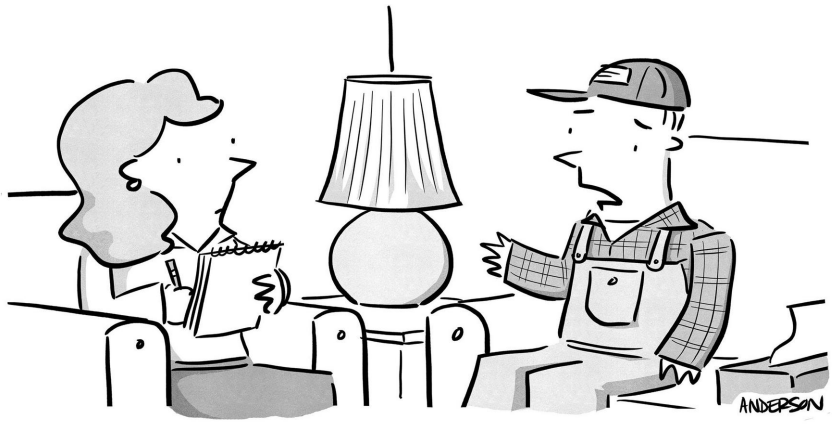
Executive roles transition from building things themselves to building teams (p. 981) that build those things. Everything important enough to be done, must be done by a team, not by a single person. One reason is the quantity of work; another is for stability, so that someone can get sick or leave the company without that part of the company grinding to a halt. Building teams is a skill set of designing organizational structures, hiring managers (not individual contributors), new methods of communication, being in command instead of in control (p. 413), prioritization (p. 221), coordination (p. 1065), and balancing the needs of the organization (p. 805) against the needs of individuals. Just because you know how to write code or design a website does not mean you know how to interview a manager or set priorities or make a decision in a meeting or effectively communicate to a large group of people.

Fixing the shortcuts, not new features only

You built an SLC (p. 101), and incurred tech debt[558] in exchange for shipping quickly. That was the correct trade-off in Explore mode, but now customers are hitting those bugs and UX issues, and hitting tech support or even hitting the little red "Cancel Subscription" link that you so desperately tried to deemphasize in a dimly-lit recess of an administrative area. Now you must pay down that debt and turn your band-aid-laden-proof-of-concept into a great, maintainable, delightful, usable product. You also need automation, test suites, CI/CD, auto-scale infrastructure, separate support prioritization and channels, and more, not only because of an ever-increasing project, but an ever-growing team of people who are coordinating their work inside it. None of that is "new features."

De-risking, not new innovation only

Now you have something to lose. Before, the company was likely to fail (p. 667) because it hadn't figured out a business model that works, but now the thing that will cause failure is operations. Tech support



"Sometimes I put all the eggs in the same basket just to feel alive."

credit 559

(p. 1503) becomes bad, the product is too buggy, the product falters under load (p. 1345), you can't keep up with marketing (p. 1027), tracking finance becomes difficult, an emergency security issue appears, taxes and legal questions arise and it is no longer cute that you're avoiding it using a self-erected vail of ignorance. You still could lose all this tomorrow; de-risking execution is now the priority, and that's a completely different mentality and skill-set from adding new features.

Solving for rare things becoming common

Before, a bug that affected 1% of customers didn't matter. Now it causes 5 support tickets every day, and mad customers who complain on Twitter. Rare things are difficult to detect and difficult to solve (p. 1345). Growth means we're rolling the dice more often every day, causing new problems to emerge. Quality becomes a priority. Requirements and processes can address it, but those concepts are new to the company, and likely counter-cultural. You hate process and requirements, remember?

Predictable processes

If you're growing fast, you're hiring a lot. The time between deciding to open a position, and having a trained, productive person in that position, is at least six months. Therefore, to know how many people you need to start hiring *today*, you have to know what the company will require in six to twelve months, and *that* means the business has to become more predictable. This is true in tech support, sales, engineering, and in standing up new departments like finance or HR or account management.

Predictability means we cannot permit disruptions, which in turn requires playbooks, documentation, measurement, processes, and other objects that maintain predictable outputs even under conditions of variable inputs. These guardrails also help new people get up to speed faster and do things your way. Your *new* way, that is.

Addressing stalls in growth

You're going to be underwater trying to satisfy customers while also hiring people, while also not having the documentation and training that those new people need, while also adding new processes, while also changing the culture to match the new "Execute" modality. Addressing all this dunks you further underwater. Meanwhile, growth can begin to stall (p. 1191) as you're lagging in new features, competitive activity, and new marketing. See that article for how to diagnose and address those stalls. You have no time to waste.

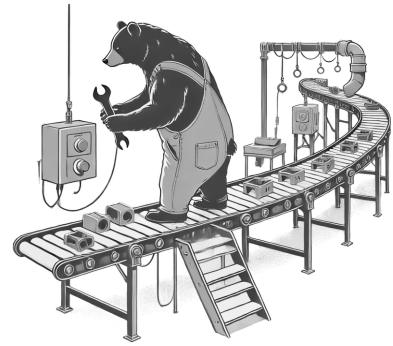
Measuring "success"

It's easy to measure success in the forms of sales leads, closed-won rate, or sprint velocity;* often the new processes and requirements of "Execute" lend themselves to measurement. This is one of the few ad-

* Personally I don't believe sprint velocity should be used as a measure of productivity or "success." It is a classic example of a metric that becomes meaningless if management is watching it; we subconsciously start over-stating point-values and it incrementally creeps up. Rather, it should be a private metric used by the team, so they can honestly measure and management themselves.

vantages of creating systems of processes and workflows. But measuring success during "Explore" can be difficult; maybe impossible. How do you measure progress when we *expect* fail at least half the time? We can protest "but we learned something," but did we?* And how do you measure *that*? Not all important things can be measured; "learning" is probably one of those things. But an explorer who wanders for years and never finds anything, is not succeeding. As a result, teams who have only known "Explore" often resist the transition to measuring the specifics (p. 645) of "Execute".

Different people, different skillsets



Consider the personality of the person (p. 569) who doesn't want a map, is happy to work hard only to throw it all away because "that assumption turned out to be wrong," and to ship everything at 50% quality and 50% completion. This all perfect for "Explore". Compare with the person who loves having a clear goal, clear constraints, then designs a way to complete the mission, and then executes that mission. This is perfect for "Execute".

But these aren't the same personalities. It's probably not the same person. Sometimes people can make the transition; sometimes they can't. This is difficult to manage, especially when those people were critical to your early success.

It's most difficult when that person is you.

---

* It can be hard to know what we learned, like Edison who apparently didn't learn anything from failures, famously quipping "all I've learned is one more thing that hasn't worked." It wouldn't have taken 1000 trials if he learned anything systematic. But even luminaries who talk about the Explore/Execute dichotomy have admitted that learning is slower in exploration than in execution, and more sporadic, and that the lack of consistent learning is one of the challenges of that modality.

And it always is.

TRANSITIONING ATTITUDE AND CULTURE

It's not just changing what you do every day, it's a fundamental change in your attitude, skillsets, and in the culture of the company.

It is realizing that **things that you're proud of** were perfect for discovering what to make, but those things are now holding you back from scaling something that's already made. It's time to be proud of something else.

This is the worst part. To let go of those old ways, the "things that got us here," and sometimes, even the people who got us here. Sometimes, you yourself don't fit anymore.

This is going to be hard.

Chapter 33: "I scratched my own itch" isn't good enough

UNDERSTANDING · PASSION · KNOW-HOW
EXECUTION · I'M THE CUSTOMER · CONSENSUS
FORTITUDE

The most common single-line origin story:

"I had the problem myself, so I built a product."

This is the standard defense mounted by new founders to justify their new ventures. But is it justified?

For a company to succeed, many conditions must be true—or true enough that they're not fatal. Those things can be formulated as a series of risks to be mitigated (p. 667) or as a list of blunders to avoid (p. 379) or as a list of questions you have to answer (p. 521). Regardless

of form, a market must exist (p. 71), at an acceptable price (p. 515) , matching customers' willingness (p. 275) and ability (p. 437) to buy.

Often left out of these lists is *you*. The company has to be *right for you*, personally. The simplest test is to ask:

Why are *you* the right person to start and grow this company?

Which brings us back to the original claim. You are the right person, because you had the problem, you were struck with a vision of how to solve it better than the alternatives, and you did.

Sounds good, but is it? Here's what you are implying by that statement:

1. I understand the problem.
2. I have an attachment and passion for the problem.
3. I know how to solve it.
4. I have the ability to execute it.
5. I understand my customers (because I am one).
6. Others have this problem, and will choose my solution.
7. I have what it takes.

The worst part is, some of those are true! Intermittent reinforcement is the strongest mechanism for conditioning. That makes it problematic: Because some are true, you can't see why some are false.

This blindness consistently results in failure. It is possible that most startups begin this way; it is definite that most of those startups fail.

Look! Smart people agree:

> "Our customers did a lot of stuff that I would never do. We think differently. We solve our problems differently. We have different needs and wants. Repeat after me: You are not your customer."
> *—Eric Ries*[560]*, creator of The Lean Startup*

> "Be a user of your own product. Make it better based on your own desires. But don't trick yourself into thinking you are your user."
> *–Evan Williams*[561]*, founder of Blogger & Twitter*

> *"If the VP of Engineering thinks the target customer is just like him/her, you're doomed. If the VP of Marketing thinks the target customer is just like him/her, you're doomed."*
> *–Cranky Product Manager*[562]

Let's pick apart the seven implications to see how to leverage "I had the problem myself" as a springboard, instead of following the usual glide-slope to demise.

"Tell me about a time when you came up with a brilliant idea that revolutionized your industry and saved your company from almost certain ruin."

credit[563]

1. I UNDERSTAND THE PROBLEM.

You do. At least, you understand *your version* of the problem. The world is large, and getting 1000 customers means solving a variety of related problems, even if you're targeting one niche.

Sometimes you really do have this broader understanding. Maybe you're a consultant and you've seen 13 variants of this problem. Or you've been engineer in this field for ten years at three different companies and you've seen everything. Or you've researched this for years as a passion. (Although, being an "expert" is another form of blindness (p. 1427).)

Either way, you're off to a great start, but you need to view "the problem" through your customers' eyes. They might see the same problem but use different language.* They might have the same type of problem but it appears in four distinct forms,** so you need to either pick some to solve, or use features and language to show how you address all four. They might need to integrate with systems you haven't head of. Their budget might be $\frac{1}{10}$th or 10x larger than yours was.

Treat this initial understanding as a starting point rather than the end-point, and interview customers purposefully (p. 239) to get the full answer.

---

* For example, Discord used to say "get a free community space," but it turned out that kids were accustomed to "setting up servers," therefore it made more sense for them to say "get a free chat server." Their community spaces are called "servers" to this day.

** For example, you've noticed that product on-boarding is a common pain-point. But Enterprise products require multi-phase, white-glove processes and training sessions, whereas small businesses might need templates and ready-access to tech support, whereas consumer products might need better self-service options and wizards.

2. I HAVE AN ATTACHMENT AND PASSION FOR THE PROBLEM.

This one is very likely to be true. You cared enough to investigate the problem and derived satisfaction from solving it. You were drawn to both problem and solution. It ticks several personal boxes, which explains why it's such a common startup origin story. I believe you.

Passion is required, because it gets you through the hard times. The pain of constant rejection—potential customers not buying, potential hires not accepting, potential investors not investing, friends and family and Twitter and Hackernews dismissing you, bad product reviews, and cancellations. The setbacks, the engineering problems that took long than you thought, the customer acquisition that isn't happening as fast as you dreamed. The pain of seeing others succeed while you struggle, especially if they're a direct competitor.*

You need the motivation to push through. You must have some personal connection, whether you love the work or the problem—ideally both. This company will be an extension of who you are (p. 1005). "I solved my own problem" is a good motivator, but you will have to be strong enough to face what is coming.

Which means doing the other things in this article.

---

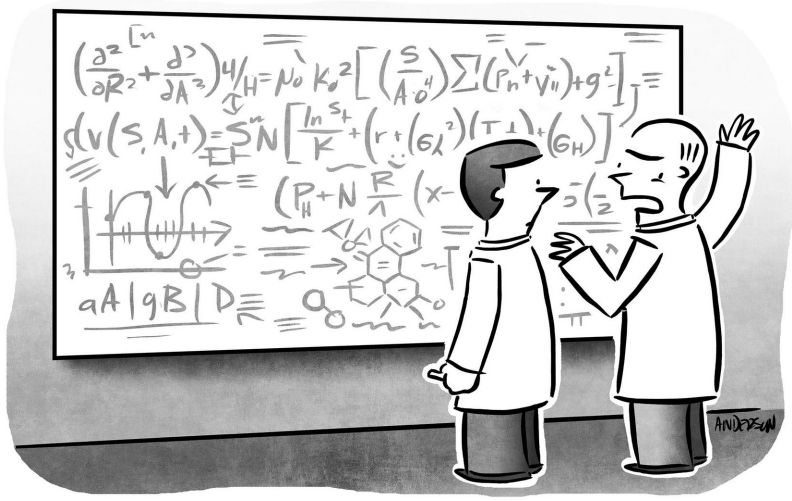
* Although, remember the day before a competitor goes out of business, all their social media and press is positive, talking about hiring, growing, new features, and happy customers. Do not believe what you read.

3. I KNOW HOW TO SOLVE IT.

You know how to solve one facet of it, where exactly one person understands it, with the exact configuration and constraints that you had.

I believe you can solve the other cases too. Most things with computers can be solved.

So, again this one is likely true. Good! There are many cases where it is not, especially in AI (p. 419) where demos look great but they immediately fail (to be good enough, often enough) in the real world.



credit564

"Stop calling it a 'head scratcher!' I *know* it's a 'head scratcher!' *Stop saying 'head scratcher!!!'*"

4. I HAVE THE ABILITY TO EXECUTE IT.

But "it" isn't the code. "It" is everything else a company must do to succeed. Most of those things isn't "writing code."

Do you have the ability to write a headline on the home page that attracts the right person in three seconds, or does it just say "_____ doesn't have to suck" because you're a lazy writer (p. 627) who has never written a high-performing hook?

Do you have the ability to get dozens of qualified leads onto the pricing page, or have you never created a single successful marketing campaign (p. 1027)? Do you have the ability to interview twenty potential customers before they buy, or do you not know how to find them (p. 683)?

There are many other things. (A list appears in the middle of this article (p. 521).) That's what "it" is that you have to execute.

Does this mean you're going to fail? No... unless you ignore all this and go back to writing code (p. 635), assuming that just because you have something working, the rest of the company will unfold naturally.

5. I UNDERSTAND MY CUSTOMERS (BECAUSE I AM ONE).

You are absolutely not like your customers.

You were so frustrated with the current solutions, decided to build your own rather than buy and implement something that existed.

So, you're like exactly none of your customers.

Your customers have day jobs, and you're lucky if this is one the top three problems (p. 479) they're experiencing right now. And yet, those three problems are what they have the budget and time to solve.

So, you have to find (p. 683) and interview (p. 239) potential customers, to learn what *they* are actually like, using your experience as the justification of your initial theory (as described in that article), iterating until (p. 845) you actually do understand your customers.

6. OTHERS HAVE THIS PROBLEM, AND WILL CHOOSE MY SOLUTION.

This fallacy is the #1 reason why companies fail when they start with "I solved the problem for myself."

I believe you identified a real problem and that you have a solution, but so many other things have to go right before there's a market that can sustain even a solo entrepreneur.

There have to be enough people who have the problem, *and* know they have the problem so that they're out searching for a solution, *and* have a budget for solving it, *and* would rather buy from you than a competitor, *and* will stick around for longer than a couple of months. The list continues; this article on finding good markets (p. 71) helps you make sure all the other pieces are in place.

While *you* might find existing solutions inadequate, potential customers might consider them "good enough," and therefore unwilling to risk switching to a new product made by a new company, that certainly lacks features, doesn't lack bugs, and might not be around in two years.

There are a hundred times more legitimate problems in the world than there are markets to buy solutions to those problems. So when you erroneously think that "where there's a problem, there's a company," you will almost always be disappointed.

7. I HAVE WHAT IT TAKES.

I have no reason to doubt you.

In fact I would encourage you: That you yourself don't know what you're capable of, until you try. Indeed, *today* you probably *don't* have what it takes, but that was true of every other successful first-time founder.

So don't give in to impostor syndrome (p. 457), but get ready to experience the gauntlet (p. 737) of Klingon Painstiks[565] that await you.

Rather than thinking it's all in the bag, think the opposite: Almost everything you'll need to succeed, you don't have yet: Knowledge of customer, market, marketing, sales, finance, skills, everything.

Yet.

Even second-time founders typically fail. I've had four successful startups, two unicorns, both bootstrapped and funded, and *still* most of my initial ideas about most things are wrong. So, everyone needs a Beginner's Mind.[566] Everyone needs to leverage their Pivot Points (p. 569) to win in their own way.

Your current idea is the spark; the right idea is still out there. You have to go get it.

I know you can do it.

Chapter 34:
The three kinds of leverage that anchor effective strategies

STRENGTH · DIFFERENTIATION · DURABILITY

"I dunno, what if we leveraged something."

"Leverage" means generating a large effect from a relatively small effort, created by riding tailwinds of natural abilities or hard-won assets, rather than fighting a battle for which you are ill-equipped.

Whatever you can do, or your company can do, that causes others to shake their heads in amazement, wondering how you accomplished so much in so little time and so few resources, is more than just a "strength." It is leverage.

Some say that great strategy (p. 489) requires unassailable, hyper-differentiated, completely unique moats (p. 761). While this is obviously *preferable*, it is unwise to ignore the more immediate and accessible forms of leverage.

Leveraging strengths is the only way to do great work. (Not "fixing weaknesses.")

> Better yet, leveraging *differentiated* strengths means you beat the competition.
>
> > Best is when that differentiation is *durable* over time.

LEVERAGING STRENGTHS

> *"How do you beat Bobby Fischer? Play him at anything but chess"*
>
> *–Warren Buffet*

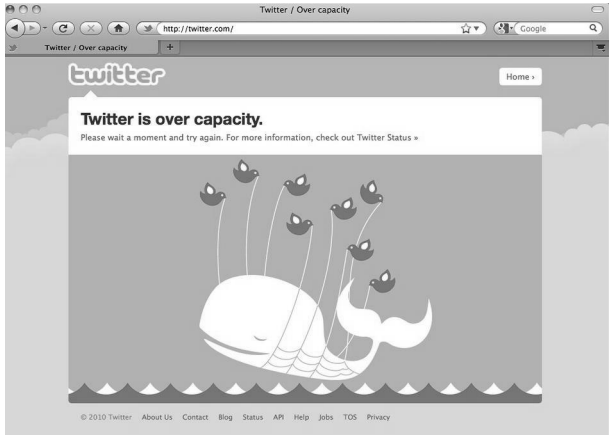
Why is it so easy to explain why a company succeeded, in just a few words?

Netflix pivoted to streaming. Snap had disappearing messages in a uniquely simple interface. The iPhone had unparalleled design. Tesla solved both performance and safety for EVs. Google's PageRank algorithm was an order of magnitude better than anything else, so it beat the other thirteen major search companies that had a head-start. Twitter's limit of 140 characters, apparently uselessly trivial, created a uniquely addictive communications pattern, and meant it could be used even on cheap devices with cheap cellular plans.

The story is always about how the company was amazing at one or two specific things. The story isn't about the 47 complex details that all added up, and isn't about a 28-page strategy document.

The story is also never about how the company "shored up weaknesses." Twitter had massive weaknesses, like constantly going off-line (Figure 1).

Fundamental issues like downtime and security flaws do need to be solved, but that's not a *strategy*, it's an operational necessity. Facebook famously made this point when they pioneered the catchy motto



credit[568]

Figure 1: Twitter Fail Whale circa 2010

credit[569]

Figure 2

"Move fast and break things," explaining that rapid innovation is more important than anything else (Figure 2).

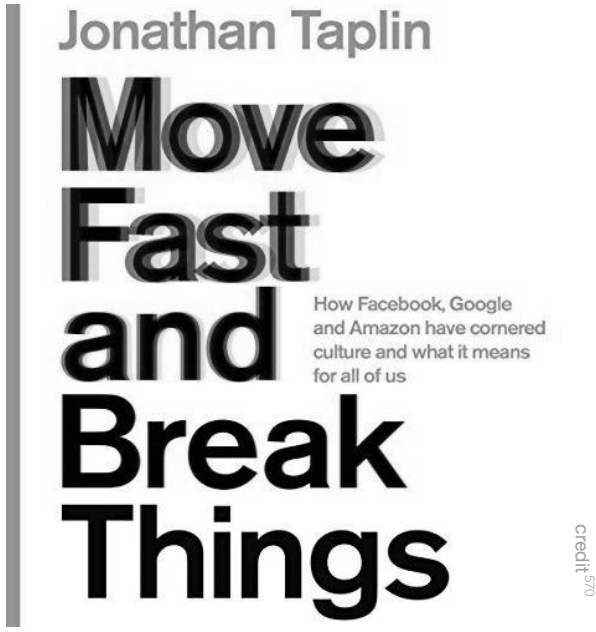


Figure 3

Books were written extolling the benefits and wisdom of this admonition (Figure 3).

Except, once you're no longer a scrappy startup, it's not smart at all. The Fail Whale is not OK. Not even for Facebook, who officially changed their motto more than a decade ago (Figure 4).

Facebook didn't defeat MySpace because of stable infrastructure or breaking things; they won because of their strengths in hypergrowth. Twitter continued to grow during the many years of Fail Whale because of its unique communications experience.

Reversing weakness is hard, painful, likely to result in something merely neutral, not great, and is at high risk of failing completely. Leveraging our innate greatness is "how we will win," and "how we will win" is the entire purpose of a strategy (p. 489).

This is why typically the best answer to "what tool/language/framework/process should I use to do _____?" is "the one you al-

credit[571]

Figure 4: No one wrote a book about the new motto. Even Zuck couldn't bring himself to write out the word "infra-structure."

ready know."* That's the one that best leverages your time—your most limited resource.

Occasionally you do need to correct a strategic weakness in order to win. Not an operational weakness like "unstable infrastructure," but one where you're constantly losing deals to a competitor, due to a major deficiency in product or positioning. Something where we need to change "who we are" in order to win.

A good strategy will explain why this risky, difficult, unnatural, expensive journey must nevertheless be undertaken, and why it's the best way to win. There must not be many; probably not more than one. You can't afford the time, people, or energy.

Better is to avoid weaknesses through decisions on target market, target personas, and product (p. 71). So if your weakness is that you

* Slack was famously built in PHP, a language derided at the time as passé. But PHP is what the founders knew, and customers don't care either way. Facebook used PHP too; when they scaled and needed more performance, they invented their own version of PHP. Expensive, yes, but companies that never get off the ground never earn the "good problems to have" of scaling.

"Flight, super strength, super speed, bullet proof, heat vision, freeze breath… And you're upset because you can't see through lead?"

credit[572]

cannot supply 24/7 customer service like some competitors, call yourself a "boutique," meaning a business where customers have incredible interactions with experts, but of course those experts are available only at limited times. If your weakness is a small team who can't build millions of features, make a product targeting users who have simple needs and appreciate simple, well-built software that "doesn't have all that crap I don't want."

Some weaknesses can be debilitating if they cross some threshold, but otherwise they should be ignored; uptime is one of them. For these types, at WP Engine we've had great success with the system of SLO management from Google's SRE Book;[573] * while written about

* You define a metric that measures the effect in question (e.g. "uptime in past 28 days"), and you define the acceptable threshold (e.g. "99.5%"). The rule is: while the metric exceeds the threshold, we do not attempt to improve it—that's like overinvesting in a weakness that will never become a strength. But if the metric drops below the threshold, the team addresses it—that solves for the "minimum acceptable weakness." By being objective and explicit about when we invest, we invest just the right amount, at the right times. Imagine a downtime incident, with emo-

infrastructure, it's a general-purpose system for managing things that should be satisfied rather than maximized (p. 887).

A cross-functional team meets quarterly to agree on the threshold; this way, when some downtime happens, and everyone is emotional and demanding that "something must be done, so this never happens again," you can go back to your rule and point

Leverage isn't just about strengths; it can also come from the business model. A common business model amongst self-funded entrepreneurs is a product with near-zero marginal cost.** Software works this way, and also content: blogs, newsletters, podcasts, videos, books, courseware. A newsletter written for 100 people takes the same effort as for 100,000, yet the latter can generate 1000x more revenue. This is why a self-funded a content company is difficult at first (all effort, no income, no compounding effects) but can be extremely profitable if it takes off (far more revenue for the same effort and almost the same costs).

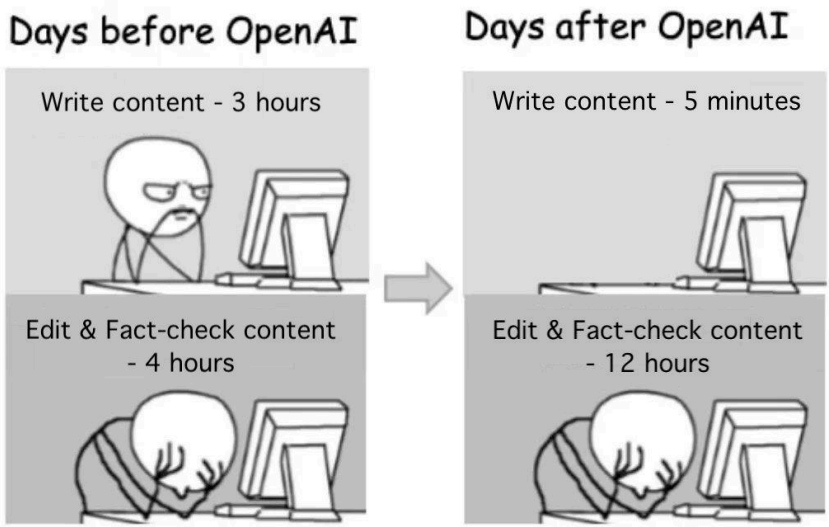
Leverage can also come from supporting tools. As of this writing, AI can generate a simple blog post, albeit with factual errors, no personal examples, and a plodding style. But, AI can *draft* ten articles, and a human can edit those many times faster than generating them from scratch, resulting in a large increase in output. Similarly, a great strategy might mandate the creation of a specific supporting system that doesn't just "make us more efficient," but makes us 10x more efficient, and therefore becomes a new form of leverage.

Passion is leverage. Your strengths are magnified through sheer will, and that same energy helps you plow through the emotional pain and drudgery of working through challenges. This is why "follow your passion" is correct but incomplete.

---

tions running and and demands that "we must invest so this never happens again." That's when you need a previously-agreed-upon decision-making rule.

** Meaning: Creating the first copy of the thing might be expensive, but selling the *next* copy of the thing costs almost nothing.



credit[574]

> *"Magic is just someone spending more time on something than anyone else might reasonably expect."*
>
> *–Raymond Joseph Teller (of Penn & Teller)*

LEVERAGING DIFFERENTIATED STRENGTHS

The kinds of strengths we've just identified are good, but your competitors have those kinds of strengths too. (You're not the only organization with "super-talented, passionate engineers," and all content companies have a leveraged business model.)

The more important strengths separate your organization from others. These are more strategic, because they help you win against the competition, not just by leveraging your time.

It's simple conceptually; these are the clichéd "differentiated competitive advantages" or "unique selling propositions." The simplicity is deceptive: it is difficult to invent differences which aren't trivial, and that also apply to most of your target audience. This difficulty is why these need to be identified in the strategy; we won't just happen across these by accident during day-to-day work.

> *"Attack where you are strong and the enemy is weak; do not attack where the enemy is strong and you are weak."*
>
> *—Sun Tzu, Art of War*

Technology and design choices can create competitive differentiation. A technical architecture will make some features and capabilities easy, and others difficult or impossible.* A design can make certain use-cases obvious and delightful, while others are annoying or impossible. There is no such thing as technology or design that makes no trade-offs. Great strategy names those trade-offs, shows how they align with the target market and target persona, and therefore how you'll

* Heroku invented the now-ubiquitous idea of the 12-factor application, but only after mandating a combination of trade-offs which many Rails developers initially hated, like requiring Bundler for package management (back when Bundler was bad), mandating PostgreSQL instead of MySQL (back with the latter was 10x more common),[575] and disallowing writing data to disk. Developers ultimately accepted these restrictions because taken together they enabled a new way to develop, test, stage, deploy, and scale applications. Today, nearly all SaaS applications follow the 12-factor principles, and Heroku cleanly won the Rails market for a decade.

exploit those trade-offs to win,* while competitors who made different trade-offs won't be able to follow, at least not at extreme cost and taking far more time just to catch up to where you already are today.

A specific type of technical differentiation is solving a particularly difficult engineering problem.** This has the advantage that, if you succeed, you might have created a moat (p. 761) (e.g. Google search, Tesla cars, OpenAI). The disadvantage is that it is likely to fail: take too long, take too much money, not actually solve the difficult puzzle, not do enough of what the customer needs. Therefore, you should be surgical about which subset of the software should be difficult: Something that creates a competitive moat *and* that leverages some strengths or assets.

Besides "big awesome features," another kind of differentiated advantage is an *insight*—when you understand an important truth while the rest of the world does not, or especially if the rest of the world confidently believes the opposite to be true.*** Their disbelief pre-

---

* But also, how those trade-offs create limitations or problems for you, that you will need to accept. You must not call it a "bug" when those limitations appear. They are the consequence of leaning into your strengths, and trying to "fix" them will take inordinate effort, they will not result in a fantastic product, they blunt the power of your strengths, and they remove time and energy that could have been spent magnifying those strengths.

** Google Wave was supposed to change everything: Replacing chat, email, word-processing, spreadsheets, file-storage, note-taking, wikis, and websites. Built as the conceptual precursor to Google Docs, it was the first product with remarkably fast multi-user simultaneous editing; you could watch people type character-by-character. The technology was fantastically difficult; no one else had it, and it would be years before anyone else did. Unfortunately it was so difficult that it failed, but this proved the rule that, had they succeeded, it would have created long-term differentiating leverage; project co-founder Lars Rasmussen said it was so difficult, that even if they were allowed to it all over again, with the same team, with all the knowledge they now possessed, it would again take years to execute, and still probably wouldn't have worked. That means the competition would have failed too.

*** "People will buy shoes over the internet." [Zappos] // "People will take rides from strangers." [Uber, Lyft] // "People want to sleep in stranger's houses." [AirBnB] // "Rockets can be reusable." [SpaceX] // "The highest-performing sedan can be battery-powered." [Tesla] // "People communicating in 140-character chunks can topple governments." [Twitter] // "Large enterprises will acquiesce to putting

vents competition. Once proved true, others will follow,**** but it's too late; the strategy has already been a success. Made famous as Peter Thiel's "Secret,"[576] these are rare but valuable insights that underpin some of the most famous and successful corporate strategies.

Unfortunately, differentiation tends to erode over time, sometimes rapidly. "Secret insights" make themselves known over time, and almost anything in software can be copied. Patents are supposed to provide protection, but what software company prevents copycats through patents?

Therefore, most valuable, strategic differentiation is of a kind which takes decades to erode, if ever.

LEVERAGING DURABLE DIFFERENTIATED STRENGTHS

> *"The most important thing to me is figuring out how big a moat there is around the business. What I love, of course, is a big castle and a big moat with piranhas and crocodiles."*
>
> *—Warren Buffett*

---

their customer data 'in the cloud' (a term we just invented)." [Salesforce] // "Open source copycats will defeat multi-billion-dollar incumbents." [RedHat]

**** Only after Salesforce convinced the world that SaaS is viable, nearly all software companies copied the model. Only after Linux defeated IBM, Sun, HP, and everyone else, there's now an open-source copycat of everything ever invented. Only after RedHat successfully monetized "free" open source software, reaching tens of billions of market-cap, did investors "open source companies" now resulting in hundreds of billions dollars of market capitalization.

The word for a durable differentiated advantage is a "moat (p. 761)." Moats often take the form: "I have something that you don't, and furthermore, you either don't have the ability to acquire it, won't invest enough to achieve it, or in the case that you try, you would have to make an incredible investment of time and energy and possibly brand adjustment to do it, which is unlikely to succeed, and even if it does succeed it will take years, by which time we've already won."

Visit this companion article about moats (p. 761) for examples, including a worked case-study.

Besides those classic moats, a special and often-overlooked form of durable differentiation is to exploit an ossified structure in a competitor, where they cannot react even if they wanted to.

- A competitor with a high cost structure *must* charge high prices, therefore a strategy of have "40% of the features at 10% of the price" leverages a point of differentiation that the competitor cannot match. This is simplistic form of Innovator's Dilemma disruption.[577]
- A competitor with a simple product and no tech support won't be able to win complex clients with complex needs, therefore a strategy of "Enterprise offering at Enterprise prices" leverages differentiation that competitor cannot match.
- A competitor who loudly, publicly espouses a set of values cannot take actions contrary to those values, therefore a strategy that espouses a contrary set of values will remain differentiated.*
- A competitor who invests in a core platform with specific trade-offs will not be able to change those trade-offs for years, if ever; therefore selecting different trade-offs, resulting in different product features (and liabilities), will remain differentiated.

* A company who values sustainable sourcing cannot price-compete with those who lack that value; conversely a sustainable-sourcing-minded consumer will buy from the former despite the expense (p. 275).

- A competitor whose profit is driven by a specific product cannot lower prices when another company decides to build that same product for free as a loss-leader that powers a different line of business:
  - Netscape (revenue from selling a browser) was killed by Microsoft with IE (loss-leader for Windows)
  - For-profit smartphone operating systems were killed by Google with Android** (loss-leader for the Google ecosystem of services)
  - Twenty proprietary Unix operating systems were killed by RedHat with Linux (loss-leader for RedHat's Enterprise Services, later sold for $34B)
  - Quoting[578] Microsoft CEO Satya Nadella on his strategic approach with Bing, "There is such margin in search, which for us is incremental. For Google it's not, they have to defend it all."

Finally, in a crowded competitive space, it's possible that no *single* thing is completely unique, much less durable. A combination of multiple things *taken together* can be both differentiated and durable, so long as the constituents form a mutually-reinforcing, cohesive structure, in which a competitor who successfully copied a fraction of the structure will still not have created a similar product.*

Consider the example of Southwest Airlines, a low-cost US airline with an incredible business model, remaining profitable even during the US terrorists attacks of September 2001 and throughout the Great Recession of 2008, across a period where every other major US airline filed for bankruptcy at least once.

Southwest doesn't have any one thing that is completely unique in the industry, and certainly nothing that a competitor couldn't copy if

** Except Apple of course; still, 87% of global handsets are Android, and Apple arguably remained because of the hardware more than the operating system.

* This idea is also explored in Blue Ocean Strategy,[579] though it originated years earlier as in the example below.

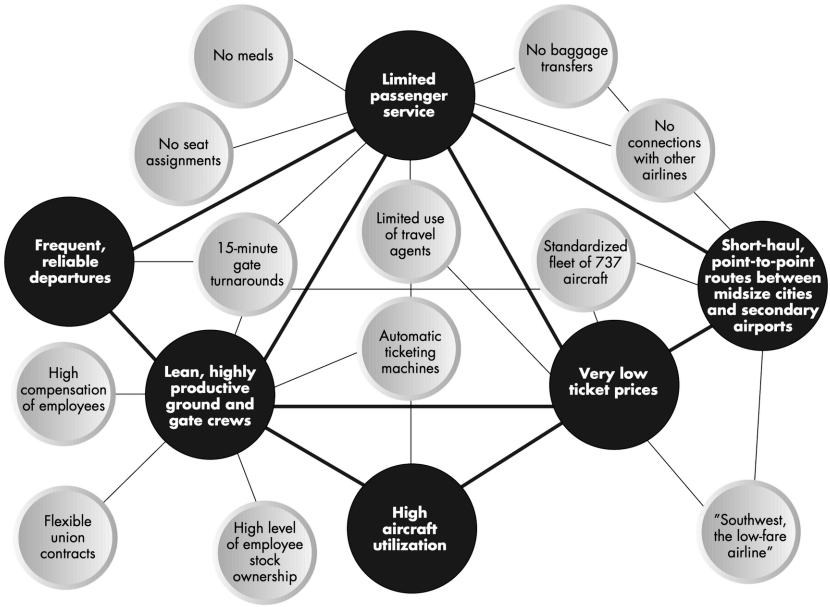


Figure 5: Michael Porter's "Activity Systems Map" for Southwest Airlines, from his 1996 article "What Is Strategy?" Harvard Business Review 74 (6): 61–78.

they wanted to, but it does have a combination of decisions, features, strengths, and even weaknesses, that together are unique, have been durable for sixty years, and explains their success. Michael Porter diagrammed this combination in 1996; his analysis is remains accurate more than twenty years later, underscoring its durability (Figure 5).

Notice how some attributes would be considered strengths by a customer (e.g. low ticket prices and frequent, reliable departures), while others are weaknesses (e.g. no meals, no baggage transfers, no connections to other airlines, no long-haul routes). The key is that all decisions support a common structure, e.g. the lack of amenities is part of what allows them to have lower prices, and short-hauls on standardized aircraft (which helps with cost-reduction) also leads to frequent, reliable departures (that customers love).

> *Strategy is a set of interrelated and powerful choices that positions the organization to win."*
>
> *–Roger L. Martin*[581]

It's easy to claim that "our peculiar set of features makes us unique," but that's not strategic unless it forms a mutually-reinforcing network, including weaknesses. It's not durable leverage unless a written strategy explains this network, so that the company can continue to reinforce the entire network with its actions and investments, including accepting the weaknesses, not treating them as "bugs to be fixed."

Otherwise, it's just a pile of features.

> *Give me a long enough lever and I shall move the world."*
>
> *–Archimedes*

Without leveraging strengths (rather than spending far more energy shoring up a weakness that still won't be great), the company will not succeed in creating something great.

Without leveraging *differentiated* strengths, the company will not surpass competitors, will have a hard time winning and keeping customers, and will have an even harder time justifying profit-generating prices.

Without leveraging *durable*, differentiated strengths, the company's success will be short-lived, differentiation will be temporary, and once again it will be reduced to out-spending on marketing or lowering prices until it is unprofitable.

A winning strategy (p. 489) explains which strengths the company will leverage, how it will side-step rather than "attack" its weaknesses, which strengths can be leveraged for differentiated sales today, and which long-term moats the company is constructing.

Chapter 35:
Legacy

MEMORY AS LEGACY · CONTRIBUTION AS LEGACY
RELEASING LEGACY · LEAVINGS

The first time you die is when your heart stops beating.

The second time you die is the last time someone speaks your name.

Or is it?

MEMORY AS LEGACY

The concept of "second death" has always been deeply human.

The ancient Egyptians believed that the afterlife persisted only as long as living people remembered one's name, and therefore built edifices that still stand with a permanence of 4000 years and counting. Jews recite the Kaddish on death's anniversary, intentionally recalling the person. Many Native Americans "keep the spirit alive" through storytelling, passed down through the generations. The ancient Greeks called it "glory" (κλέος); Achilles is given the choice between a long, uneventful life or a short life with enduring fame, and chooses the latter. The Japanese Obon Festival and the Mexican/Aztec Día de los Muertos reaffirm continued second lives through remembrance. The Korean tradition of Jesa invites the departed with food and bowing, and the Chinese Qingming Festival (清明节) brings honors and remembrance to the tombs of loved ones.

In modern times, the rich perpetuate their names on charitable foundations and carved into university building cornerstones. Discoverers' names endure in theorems (Pythagorean) and particles (Higgs) and frameworks (Myers-Briggs) and conquered lands (Sandwich Islands).

But is this the best way to live beyond death—affixing our name to a thing?

Recently I've become enamored with a different concept of legacy:

The on-going impact of ideas.

CONTRIBUTION AS LEGACY

The authors of religious texts are largely unnamed, and yet their stories, lessons, laws, and ethics have determined the world view of tens of billions of humans, and will continue to do so.

If Charles Darwin had written his book anonymously, his ideas would still have echoed through science. Newton "stood on the shoulders of giants," even if we don't bother to enumerate those giants ourselves. If we erased all names from all discoveries, they would not cease to be discoveries, and their effect on us would not diminish.

As Juliet observed, a rose by any other name would smell as sweet, and her world would have been happier had it not been poisoned by a name. The plot of "Romeo and Juliet" predates Shakespeare by thousands of years in the forms of Pyramus and Thisbe, Layla and Majnun, Tristan and Isolde, 梁山伯与祝英台 and surely many others. Though the ultimate origin is lost in time—and might not even be singular—ideas can echo wide (in geography) and deep (in time).

Like water waves, ideas propagate without reference to the originator. The creator lobs an idea-stone into the lake of humanity. Waves of influence move in all directions, potentially infinitely far across time and space, the original stone forgotten, or never known. Waves fizzle out without support, but they grow if whipped up by winds of repetition, and they grow when they inspire other creators, remixing additional ideas into new stones.

Most waves become indiscernible on the tumultuous surface, but they were undeniably part of the force eventually crashing on a distant shore of a distant mind of a distant human.

It's happening to me, now. People are repeating my SLC (p. 101), sometimes outright plagiarizing the original.[582] Some consultants[583] and agencies[584] present SLC as their own idea. And another[585] and another[586] and another[587] and another[588] and another[589] and another[590] and another[591] and another[592] and another[593] and another.[594] Even video.[595]

When I first discovered these, I was indignant, maybe even bitter, maybe even angry. Who wouldn't be? But these are the ripples on the lake. These are the winds of repetition with no reference to the originator. If the ripples are the legacy, then they can be met instead with gratitude and pride.

RELEASING LEGACY

What does it mean for all of us, today?

It means we should worry less about "content" for the sake of "eyeballs" and more about creating ideas that are worth repeating—ideas that are so important and true and compelling and repeatable that people *wish* to repeat them, even if our names aren't attached to the broadcast and our tweets aren't the ones being retweeted. We should worry less about the attention we bring to ourselves, and more about how we can positively influence others, how we can help others become (p. 751) better versions of who they already are.

This is all easy for *me* to say. I have a modicum of fame from writing about 25 years of building companies. This article was broadcast to 60,000 newsletter subscribers.[596] It's also easy for me to say, having made enough money through building those companies that I was never compelled to monetize content. So sure, *I* can say "it's not about the fame nor money." A rare luxury.

Of course attribution is not without value. Artists should be paid for their work. Fame and money matters while we're alive, and these are not trivial things. Proper attribution is ethical. Plagiarism is theft.

Still, for all of us, in addition to accumulating fame and money it's useful to ask: And what else?

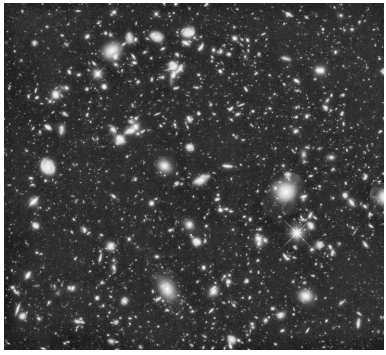
WHAT STONES WILL YOU DROP?

Which of your ideas deserve repetition? Which of others' ideas deserve repetition, with your own take, your own style, your own exposition? What will you do with your influence? What are the behaviors you wish to encourage in others? What are the lessons you wish humanity would heed?

I've been unwittingly answering those questions on this site for 18 years, without labelling the ideas with my name. Hence not only SLC (p. 101), but Binstack (p. 603) and Needs Stack (p. 259) and Talk/Walk (p. 949) and Fairytale Strategic Planning (p. 1065) and Brittle Points (p. 905) and Pivot Points (p. 569) bearing self-descriptive names, or crediting another in Fermi Estimation (p. 171), or in cases

of my Roadmap for PMF (p. 9) and Love/Skill/Need (p. 399) and Love/Utility/Coercion (p. 275) and "Is there a problem (p. 71)," and Unpredictability (p. 193), remaining simply untitled.

You must drop some stones, because the alternative is a nihilistic descent, realizing your life is a flicker between the 13,770,000,000 years of near-nothingness that came before you and the far longer expanse after you die, most of which will be a featureless soup of maximum-entropy minimum-energy photons. And even during this impossibly improbable blip, you're just a speck on a soggy marble that is a speck in a star system that is one of billions in a galaxy that is one of billions in a universe 880,000,000,000,000,000,000,000,000 meters wide.



credit[597]

And therefore we should live only for today, and only for ourselves, because to an extremely high degree of precision, we are nothing, and nothing matters.

I can't argue with those numbers, but I don't have to make that conclusion.

Because it *does* matter. To be human is to defy the uncaring universe by caring deeply anyway—caring for our loved ones, caring about our creations, and caring about our legacy.

This is perhaps the meaning of life: To discover who we are, to be driven by what we were meant to do, to create beloved and useful things, and to love and help others.

My dad died 20 years ago while I was holding his hand.

I had a vivid dream a few weeks after his death. He's looking directly into my eyes. His face looks like it did when I was growing up, not the way it did in those final moments. He's not speaking. Somehow I hear the message regardless. I don't mean a ghost is communicating, I mean my brain is processing something—something deeply important—just as thoughts intrude upon meditation. The dream intruded upon me, carrying wisdom that some part of my brain already understood. The wisdom was that although someone is dead, as long as you remember them, as long as their words or ideals are reflected in what you say, what you do, what you value, then that part of them is literally and physically living on, in you.

And you to the next.

Chapter 36:
Pivot Points

PIVOT POINTS · PERSONAL · COMPANY
APPLICATIONS

"This?! Oh please! It's not even mine!"

Gallup® StrengthsFinder[599]® is one of those registered® trademark® bedazzled® personality tests claiming to be *uniquely* insightful, honed over *decades* of scientific validation, in contrast to all those *other* personality tests who only *claim* to be data-backed, but are *actually* horoscopes recast in corporate lingo,[600] that are thinly veiled excuses for consultants to charge you for assessments.

Oh, by the way, your Official® Gallup® StrengthsFinder® Assessment® costs[601] $59.99.

But OK, for years at WP Engine every new employee takes the StrengthsFinder® assessment as they on-board, and they're treated to their "Top 5 Strengths". We've done it thousands of times, and people like it. I like it too, to be fair. It's fun to feel important enough to warrant analysis.

My third-strongest strength is "Competitive", which means what you think it means: I always want to win.* Specifically, I want to beat other people in order to win.**

Is "Competitive" a *strength* though?

> **Yes**, if I'm in a sales call. Yes, if I'm the last interview for a candidate who has other offers. Yes, if the person I'm trying to beat is "my past self," generating motivation to become a better version of myself (p. 1433) without overflowing into feeling that I'm unworthy (p. 457).
>
> **No**, if I'm in a team meeting where the goal is to help each other (p. 1017) rather than to win the argument. No, if it sucks the joy out of games and hobbies. No, if I base my self-worth (p. 1407) on (my false perception of) other people's accomplishments.

So if it's unclear whether it's a "strength," what *is* it, exactly?

---

* The part of Capitalism that is good.

** The part of Capitalism that is bad.

"My greatest weakness? I'd have to say depth perception."

credit[602]

THE RIGHT ANALYSIS: PIVOT POINTS

"Competitive" is a fact-of-the-matter of my personality. It's a thing that exists, outside of judgement. I call this non-judgmental fact a **Pivot Point** because of how you can use them to win at life and business. The rest of this article explains how to do that.

Pivot Points are strengths in some situations, hindrances in others, and irrelevant in still others. This is why I don't like the ideas of "strengths" and "weaknesses" generally—whether in personality tests or SWOTs[603] or other tools of strategy and planning. Those frameworks imply that we already know the context for evaluation, but often we don't. Pivot Points are neither intrinsically good nor bad, they just *are*.

But why the word "Pivot"?

Because you're "stuck there" like your "pivot foot" in games like basketball and lacrosse and Ultimate Frisbee. Your other foot—and the rest of your body—is free to move anywhere, subject to that constraint. This is the correct metaphor for using Pivot Points in life and strategy. These are the constraints you build around. For example if you hate managing people (Pivot Point), you shouldn't become a manager, nor should you make a company that will eventually require a team, or you shouldn't be the CEO of that company.

The other reason is that Pivot Points can change, but not often, and not capriciously. Sometimes you pick up your foot, run somewhere else, and establish a new pivot; this is investing in a new skill, or learning a new industry, or overcoming something that's a weakness relative to your goals. Short-term planning should assume Pivot Points are fixed, because in the short term, they are. Long-term planning, especially when deciding where to investing your time and money (p. 867), can ask: What new Pivot Points do we want?

This also neatly matches the "pivot" concept from Lean Startup (p. 1425): When a company realizes that one aspect of the business is fatally broken (p. 193), yet other aspects are insightfully correct. You don't throw everything away; you're realizing which things are correct or immutable—a.k.a. Pivot Points—and then pivoting the rest of the company around those things.

So let's see how to find your personal Pivot Points (for your life), as well as those of your company (for strategy), and then how to exploit them to win.

FINDING YOUR PERSONAL PIVOT POINTS

"Know thyself!"* We *ought* to know ourselves, but we're too close to the problem. You're biased, you can't get outside your own head, you have baggage (p. 1055), and few of us are naturally talented at honest introspection.

What drives you

These prompts will help you uncover who you are.

- Even when I was a kid, I would _____, and I still find myself drawn to it.
- Lately, I love it whenever _____ (which would surprise my ten-years-ago self).
- If I could go back in time, I would tell myself to worry less about _____ and more about _____.
- When I'm on an extended vacation, I get itchy to _____; I just can't help it.
- My parents/friends always laugh when I start talking about _____ because I get so excited I can't stop talking about it.
- My parents always said I would be a _____ because even when I was just six years old...
- Whenever I _____, I get lost in the work, and feel energized (not exhausted!) when the work ends.
- If I could go (back) to college, I would get a degree in _____.

---

* Inscribed at the Temple of Apollo at Delphi, independently opined by many people throughout history, Thales gave this as the answer to the question: What is the most difficult thing? When asked what is the easiest thing, he answered: To give advice. Which is what I do in these articles. The truth hurts!

- The last project I *really* enjoyed, and wish I could keep working on, was _____.
- I was surprised how much my peers praised my work when I _____; maybe I'm better at that than I thought.
- Recently I was totally immersed, engaged, excited, and happy while doing _____.
- I asked a few people who know me, and who I trust to be thoughtful and observant, and they said my special strengths are _____.

What hinders you

Sometimes it's easier to avoid something you hate, then to construct a situation where you're constantly doing something you love (always invert (p. 379)):

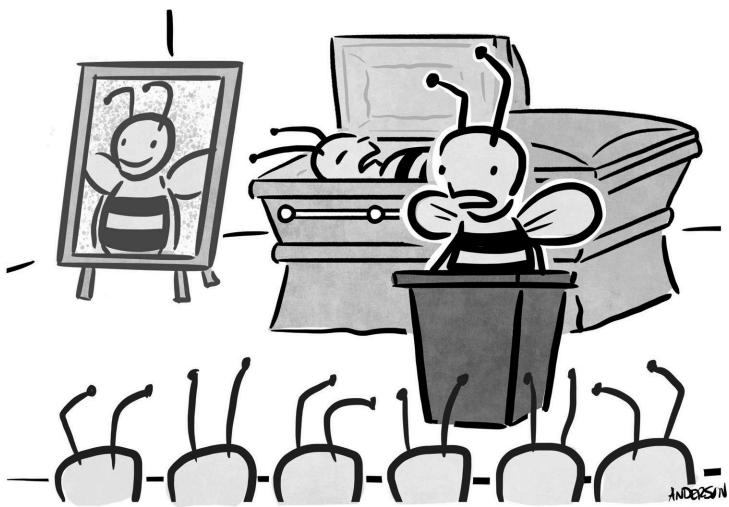
- When I'm faced with _____ I physically feel the "pit of my stomach" falling.
- The last time I had to _____, I did an embarrassingly minimal job, because I couldn't bring myself to do better.
- I have an intense dread of any meeting where we _____.
- If my job starts requiring me to _____ with even 10% of my time, I would at least think about changing jobs.
- When I'm faced with _____ I procrastinate so much, I do chores that I dislike and normally avoid.
- If I'm being honest, although I would really love to be great at _____, the fact is I'm just never going to be good at it.
- Whenever I _____ all day long, I know I do great work, but I'm absolutely exhausted; the rest of the night will be vegging out of the sofa with mindless entertainment.
- I know I'm supposed to like / do _____, but the truth is I never get excited about it.

You should be honest, even if you think "society" doesn't like your answers. There's nothing wrong with the drive to make money, or to become famous, or to prove a point.

> Desire to seem clever, to be talked about, to be remembered after death, to get your own back on the grown-ups who snubbed you in childhood, etc., etc. It is humbug to pretend this is not a motive, and a strong one. Writers share this characteristic with scientists, artists, politicians, lawyers, soldiers, successful businessmen—in short, with the whole top crust of humanity.
> —George Orwell, Why I Write[604]

Outside-in

Ask other people about your strengths and weaknesses—yes, that language, because it helps others generate answers, even though you're



"He died doing what he loved - stinging some random guy one time."

credit[605]

going to untwist their observations into non-judgmental Pivot Points. Unfortunately they're rarely fully honest, even when anonymous.

My favorite reference-check interview question can help. Candidates refer you to friendlies who are primed to say nice things, so there's no value in their sanitized recommendations. Instead, I ask the following:

> What is the ideal scenario, under which this person thrives?
>
> That is: Construct the absolutely perfect set of circumstances—subject matter, goal, team (or lack thereof), direction (or lack thereof), incentives, etc.—where this person absolutely kills it, is efficient, is productive, is extremely happy, and makes everyone around them happy.

The answer generates Pivot Points. If they say "Oh yeah, you need to give them a goal because they want to know what 'success' looks like, but then get out of their way because they're really creative and they explore quickly and you don't want to squash that," then you extract Pivot Points like "goal-oriented" and "craves exploration". That means working in an R&D group might be a bad fit (not goal-oriented), as would working on a mature product (needs execution, not exploration), but an early-stage startup would be great (clear goal, and exploration is mandatory and valued).

Also ask the opposite:

> Construct the scenario where they die inside, mess everything up, and piss everyone off.

It's funny to ask it and easy for them to answer, but doesn't feel like you're being negative about the person; after all, we all have a personal hell. More Pivot Points.

It's not easy to know yourself, and "yourself" changes over time (p. 1433). But it's worthwhile to identify major Pivot Points, because pivoting around them (or deciding when and where to invest in altering them) is how you construct a fulfilling life (p. 399).

Now we'll do this for a product or company.

FINDING YOUR COMPANY'S PIVOT POINTS

"Knowing thyself" is hard enough for an individual; collectively we're even more complex, and each person holds different knowledge about different areas. We can't ask our team to just "write down what we are like." Here's what to do.

Suppose you knew nothing about the inner workings of your business. Pretend you are an outside consultant, hired to reverse-engineer every aspect—not what the people inside the business *think* it is like, *think* are its strengths and weaknesses, but what's *actually* going on. You'll be sussing out the decisions that led to their strategy, even if they never consciously made those decisions or wrote down that strategy. You job is to write down "what is."

You're allowed to observe behaviors and results, but not allowed to interrogate people to understand *why* they took those actions. Indeed, it's important that you *don't* interview people, because they will unwittingly rationalize, or even mount a defense. We're discovering, not judging.

The generative procedure starts with write-storming* observations about **what is special or peculiar about the product and company.**

Precipitation

You can't ask people to list "things about us that are special and important"—it's too vague, and won't be comprehensive. Instead use specific, evocative prompts:

Undeniable comparative strength
What specific things are customers buying, when they choose us despite our many foibles (p. 891)? What does our company do so well,[609] even the competition admits it during their own sales calls, because otherwise they'd lose credibility? What is our team capable of (p. 1271), that most other teams are not?

(Whether it is difficult for others to copy or not, whether customers or competitors seem to care about it or not.)

Consistent complaint
What complaints do our customers have, that we've heard so many times in so many channels that we don't even need data to know it is true? On what point does a competitor instantly win, because we have no defense against that claim? What is so problematic that people leave us for this reason alone, even when they're otherwise happy, even when they rate us 10/10 on NPS and apologize as they cancel?

(Whether it's smart for us to react or not, whether it's intrinsic to our product or relative to a competitor.)

---

* Write-storming[606] is brainstorming in which participants write ideas in isolation, then bring them together for deduplication and enhancement. This eliminates the bias towards the fastest, loudest voice, supports people whose brains need more time to ponder, takes advantage of background-processing (i.e. "I thought of something in the shower"), and more fully includes non-native speakers who will be much more precise and compelling when given time to translate. "I was shocked to find there's not a single published study in which a face-to-face brainstorming group outperforms a [write-storming] group"—Leigh Thompson, *Creative Conspiracy*[607]. Also, recent research[608] suggests that people are less creative in generating ideas over video-conferencing versus in-person, but there is no such difference in the quality of *selecting* good ideas from a list. Therefore, the optimal process appears to be write-storming to maximize creativity, then meeting up for synthesizing.

Customers advocate for
When our customers brag about us on social media or in private meetings, when it's genuine advocacy, what do they highlight? What positives do even disgruntled employees begrudgingly admit on Glassdoor? What concessions do one-star reviewers make?

(Whether we think they're exaggerating or not, whether it's what the majority of customers would agree with or not, even if the NPS-0 customers disagree, even if customers in their exit interviews disagree.)

Best customers
What are the characteristics of our best customers: The ones who pay the most, never call tech support, score 10/10 on NPS, never leave us, upgrade consistently, advocate for us externally. More specifically: What characteristics are prevalent that aren't shared by other customers?

(Whether that's a large market or not, whether it's shared by only 30% of them, so long as it's shared by <10% of the rest of the population.)

Proud of
What about our product or team are we especially proud of,[610] whether because we consider it to be great workmanship, or because it reflects "who we are," or because it was difficult to do, or because it's just fun? Or the opposite: What do you absolutely hate that some other company does?

(Whether customers agree or not, whether there's data to support that feeling or not, whether it's a competitive advantage or not.)

Talking & Walking
We say we're great at certain things on our website and sales material. How much of that is true? What else are we great at, that we don't brag about? We say certain things because we know that our customers desire that quality, and we at aspire to be great at it. Use the Talk/Walk Framework (p. 949).

(Whether or not this directly harms sales or retention.)

Clear and present existential threats
What is either happening now, or has at least a 70% chance (p. 997) of happening in the next few years, which would seriously disrupt our business? Something that would dramatically tank sales, where customers might start cancelling in droves, maybe even end the entire business.

(Whether there's anything we can do about it or not, whether it's our own fault or not. But hold to the "70%-100% chance" requirement so that there's very few if any; otherwise, every company has twenty theoretical disaster scenarios.)

Structural attributes
What does our technical architecture or human organizational structure specifically make possible, easy, or natural? Or the opposite: unstable, manual, too difficult, unscalable (p. 773)?

(Whether we think that's a "good thing" or not, whether competitors are similar in that respect or not.)

Philosophy & Values
Things you believe in so much, you'll honor them even if it loses you customers, even if it loses you money, even if it slows you down, even if you have to fire people because of it. Because this is who you are (p. 827). What do you believe in your gut, about what it means to do great work, what kind of organization you want to build, what impact you want to make in the world. Example: Is "great design" absolutely critical for success, at least the kind of success you want to be a part of, or is "design" overrated (p. 853), nice-to-have but a functional product and superior sales will beat it every time? Example: Is "ship fast with bugs"[611] the best policy, because you learn the fastest and therefore get to the truth and product/market fit the fastest, or is "ship fewer things with high quality" the best policy, because you win lifelong loyal customers with craftsmanship and care, not by exploiting them as unwitting lab rats (p. 101)?

(Whether lots of other people agree or not, whether there are lots of successful examples to point to or not, whether this is the "in vogue" thing to believe or not.)

"Is this about the wheel again, Og? Because if it is, I swear to God…"

credit[612]

Great ideas
What ideas keep recurring, typically because of an ingrained sense of conviction stemming from one of the areas above, i.e. we *would* be proud of it, we think it *would* become a comparative strength, we think customers *would* become advocates because of it.

(Whether customers are actually asking for it or not, whether or not we have objective evidence that it really is a great idea.)

Condensation

You'll find this exercise unearths all sorts of things you can act on. For our current purpose, we distill them into fewer, clearer Pivot Points.

You do that with the following workshop:

1. **Rotate through observations**
   Go around the room, asking each person for *just one* of their better observations. We'll get to all of their ideas, so don't stress

about which one is "best," just pick a good one.* Together the group processes one observation at a time.

2. **Clarify**
   Everyone asks clarifying questions—not arguments—to ensure we understand "what it is." Boil it down to a few words so it can go on the board (which is a whiteboard, or Miro, or whatever you like). Use one color for the observations.
3. **Convert to Pivot Points**
   The group asks: What Pivot Point does this translate into? For example, "Customers post screenshots on Twitter of long wait-times in tech support" becomes "Slow tech support." Or "Customers brag about us by posting side-by-side screenshots of our product next to a competitor" might become "Delightful, remarkable design." Write the Pivot Points in their own color, and place the observation next to it. Themes will emerge as some Pivot Points will accrete many validating observations. And the other direction: if there are multiple target Pivot Points for a single observation, duplicate the observation card.
4. **Don't evaluate; don't act**
   It's tempting to jump to a conclusion like "Slow support means we should hire more people," but fast support might not be of strategic importance, so maybe you should deemphasize support from marketing and sales materials instead of "fixing it." And even if you decide you *do* need to fix it, should you really hire more people, or should you instead write better documentation, start using AI support chatbots, switch from tickets to chat, improve the UX of the product to prevent common questions, change what segment of the market you're marketing to, or what? "Now" is not the time to answer that question.

---

* This is a facilitation trick I learned from Jonathan Slain.[613] It ensures everyone gets time and attention, not dominated by the people who jumped at the chance to go first and unload dozens of items.

5. **Merge Pivot Points as you go**
   It's better to have fewer, stronger Pivot Points. A smaller set is easier to hold in mind, easier to act on, easier to set in stone. But beware the trap where you make them so general that they cease to be useful. For example, "We love our customers" is too generic. Use the Opposite Test (p. 1589) to make sure you haven't gone over the line. When in doubt, don't merge; being specific and actionable is more important than quantity. And definitely do not have a target quantity (p. 1249) like 3 or 10. There are as many as there are.
6. **Loop until exhaustion**
   As you repeat the loop around the room, you start to skip people who are out of items, until everyone's list is exhausted (and so are they).*

The process will throw off lots of ideas, follow-ups, and potential actions. Collect those during the process, and visibly write them down elsewhere so people know their ideas aren't lost. But stay on-task.

APPLYING PIVOT POINTS

Great strategy (p. 489) and a fulfilling life (p. 399) starts with honoring Pivot Points as facts-of-the-matter that you construct strategies and careers around. There are articles on this site about how to do that, and there will be more in future.

---

* The process moves slowly at first, as people get used to the system and talk too much about each item. People will look at how many more items remain on their own list, and will worry this will take eight hours to complete. But it never does; you accelerate naturally (or the facilitator can start pressing for brevity), and people realize some of their items aren't that important relative to what's already up there, or are duplicative, and suddenly it's all finished.

Here's some direct ways you can use Pivot Points create better product strategies and a better life:

1. How does our current strategy, or a proposed strategy, map to our Pivot Points? Where is it reinforced (therefore we should go deeper) or contradictory (therefore we should pivot)?
2. How do we do even more things that leverage multiple Pivot Points? When we do that, not only are we perfectly suited to it, they are automatically differentiating in the competitive market, because others would have trouble copying us, because their Pivot Points are different from ours.
3. What changes to our strategy would design around the negative consequences of our Pivot Points, so we're not in our own way, or playing a game that others are better suited to win?
4. Given the Pivot Points we identified, what additional ones must also be true, either as a downstream consequence, or because any self-consistent strategy would have to include them? Add those to the list. Yes this is recursive and it's turtles all the way down; that's what a "self-consistent, mutually-reinforcing" strategy looks like.
5. Is there a singular concept[614] leveraging/avoiding multiple Pivot Points, around which we could build our entire positioning and strategy?
6. What type of customer—demographic, firmographic, job-to-be-done, budget-size, location—would find our Pivot Points refreshing and attractive? Is that our current ICP (p. 317), or how should we adjust it?
7. Are we making promises we can't keep? Are there promises we should be making because we'll naturally win there? In short: Does our marketing match our behavior (p. 949), and what level of the Needs Stack (p. 259) should we be operating at?
8. Are there contradictions in our Pivot Points? For example, do we want to deliver "Instant 24/7 support" but also "Can work only 4 hours/day" because of a day job and a family (p. 901)? Some contradictions must be resolved by changing the Pivot Points, but

sometimes we can synthesize new ideas that make everything work after all; these are often our most valuable and differentiating ideas. Here's how to navigate balance vs choice vs synthesis (p. 589).

9. What if one of our Pivot Points are wrong, i.e. it's not the truth? Can we challenge each one, making sure we *know* it is true? If it weren't true, what else would change? Should we actually make that change?

Here are specific examples (p. 295) of how younger startups can defeat large incumbents, when the startup leverages their own Pivot Points while designing around the Pivot Points of the incumbent. And how those same Pivot Points are different with AI (p. 419).

In my case, being Competitive lends itself to entering a large, existing market with stagnant competitors, with a strategy of "build a 10x better product[615] and then win in marketing and sales." On the other hand, someone who hates the idea of having to "win" at "sales" could build a niche product, simple enough to explain and advertise that it doesn't need a sales team, with a brand voice of a one-person boutique that makes a perfect, specialized product and IYKYK. Or someone who loves people (Pivot Point) but hates sales (Pivot Point) could start a consulting company where they forge long-lasting relationships with clients through on-going work like taking over an entire department of the company (e.g. marketing, support, engineering, or finance). All of these are wonderful strategies, so long as they match the founder's Pivot Points, and the needs of the chosen market (p. 71).

Some people call Pivot Points "enabling constraints".[616] The theory is that constraints helpfully tame complexity, and then increase[617] creativity.[618] Which they do; one might say the definition of "design" is "art + constraint".

I don't like the "enabling constraint" phrase, however, because people get hung up on constraint. "Constraint" sounds limiting, and indeed it is, even if it also triggers new and better ideas. It's not obvious how being "limited" is also "enabling". Whereas, "Pivot Points"

credit[619]

"It's a transformative piece touching on themes of joy, desire, loss, and fourth quarter earnings."

are facts with all manner of consequences, that we should leverage or design around.

Finally, although you can invest to change Pivot Points, beware. Some things might be too difficult to change. Some things might be too expensive to be worth changing. There's a risk that the change isn't successful. Change disrupts the organization (p. 1299). Sometimes employees get mad and leave. Sometimes customers get mad and leave. Sometimes people can't change. Sometimes they shouldn't.

Change Pivot Points only when the upside is so enormous, that even if you achieve only 30% of the transformation you were hoping for, it was still worthwhile. And don't do too many at once. Maybe just one, whether for yourself, or the company.

As Tim Ferris describes it, when anyone asks whether they ought to write a book:

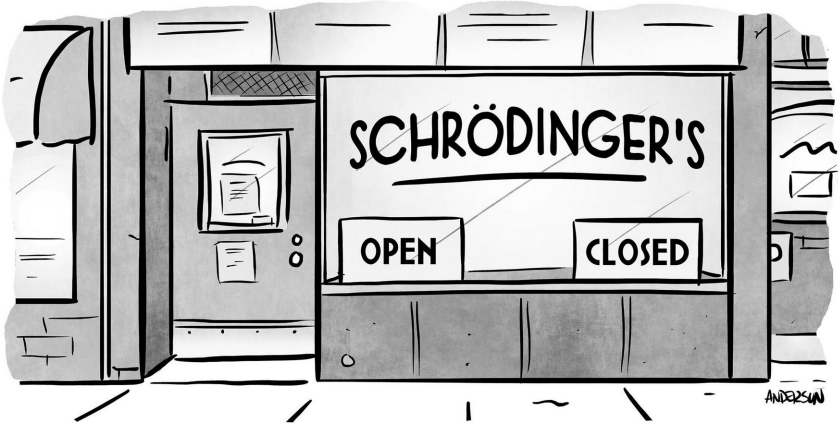
> If the rewards are 1/4 of what you think they will be, and it takes 2x longer than you think it will take, is it still a *no-brainer* to do it? Not just "good" but a "no-brainer," like you almost have to do it?

When you design a life or a strategy around Pivot Points, it's magical. It means you're "following your calling." It means you'll be more efficient, more effective, produce better quality, compete better against those who don't share your Pivot Points, and have a lot more fun. It's fun to do what you're meant to be doing.

Life is short and companies are difficult. Don't fight Pivot Points. Embrace them.

Chapter 37:
“It’s a Balance” isn’t always the answer

“IT’S A BALANCE” · “IT’S A CHOICE”
“IT’S A CHOICE… TO A LIMIT” · “WHY NOT BOTH?”



Faced with two conflicting choices, what should you do?

Ironically there are several answers, which are themselves conflicting:

"It's a balance"
After his enlightenment, the first teaching of the Buddha was to solve contradictions through "The Middle Way" (Majjhimāpaṭipadā). Aristotle instructs the same in *Nicomachean Ethics*, observing that one should be neither cowardly nor foolhardy; rather one should be courageous—facing and understanding risks, while also overcoming fears. Therefore, avoid extremes, and find the balance.

"It's a choice"
Only by making a clear and extreme choice can you enjoy the full benefits of that choice. Otherwise your energy and output are diluted into failure. If a product could either be inexpensive and target small businesses, or expensive and target large enterprises, the correct answer is not a middle-price targeting the mid-market. Strategy requires making decisions (p. 489), not compromise.

"Why not both?"
When faced with solving the global energy crisis, picking between affordable dirty energy and clean expensive energy, surely the answer is to invent energy sources that are both clean and affordable. Though it's an over-used example, Tesla's cars were unique because they were both all-electric and high-performance (and maximally safe)—factors which others assumed or even insisted were impossible to achieve simultaneously.

Since there are multiple approaches, we need a framework for analyzing the decision, and deciding which type of resolution is appropriate.

"IT'S A BALANCE"

This is the correct approach when both extremes are bad.

In the Aristotelian example, the poles are both poor choices—we should neither be too scared to act, nor oblivious to risks and reality. That means we want to distance ourselves from both, not "take the best of both." Presumably there is something in-between that reduces the negatives of each, thereby achieving a solution that is superior to either extreme.

You could visualize this as a curve you're trying to maximize (Figure 1).

"Work vs Rest" is common example. Athletes who overtrain don't achieve peak performance. Athletes with natural talent who don't train will never become great. Athletes therefore need a system that intelligently alternates training and rest.

We're tempted to call this a "balance," but I dislike that word because it can imply "equal parts of each," which is rarely optimal. It is not only the *right* amount of each, but also combined with a system that forms an integrated solution. So for an Olympic athlete it's not "equal training and rest," but a plan that results in peak performance

Figure 1

on the critical day when they will contest the gold medal. The "right system" will vary by person, by goals, and by circumstance.

"Zero-sum games (p. 295)" are another class of puzzle that falls into this domain—where more of one thing necessarily means less of the other. Time and money are often like this; when you spend time or money on something, it means you're not spending it on something else. So if the question is whether a solo founder should be spending time on product or marketing—making stuff or selling stuff—the answer has to be something like (p. 9) "a balance," but that doesn't mean splitting time equally. It depends on what the company needs more right now: more new customers, or retaining existing customers. Or stated in the negative: Which is the biggest problem right now: that customers are rejecting the product, or that not enough potential customers are trying it?

"IT'S A CHOICE"

This is the correct approach when both extremes are rational, yet contradictory.

A example facing all B2B companies at their inception: Should you target the low end or the high end of the market?* The world contains a thousand times more small companies than large, but small company budgets are also a thousand times smaller. When you're a company like WP Engine**—a platform for "websites," needed by every company of every size—it is tempting to say "we'll serve everyone," as this maximizes the size of the potential market. We'll be simple enough for

* Of course it's typically more segmented than this, but let's be reductive for the sake of example.

** I founded in 2010, now a unicorn with 1200 employees, profitable and growing, serving many market segments.

small companies, but with mature compliance and features for large companies.

Tempting, but wrong. A new startup must find its Carol—it's ICP—then focus (p. 317) positioning, marketing, brand, and most of all, product, on that one segment. The linked article explains why.*

Hopefully it's obvious that you shouldn't pull an Aristotle, saying "We'll make a product for the mid-market."[621] While that may indeed be a good segment, you would make this decision by analyzing the best segment for your company, not because mid-sized companies are a "balance" between small and large.

When the extremes are bad you seek a balance, but when the extremes are both rational choices, it is not a *balance* but rather a *choice*, and choose you must. Trying to find a "balance" means "failing to make a decision," which is at least suboptimal, and probably downright bad. Great strategy requires making clear, firm decisions. (p. 489)

The diagram collapses to a choice (Figure 2).

This is often the case with strategic decisions. Consider the choice between building a business that is profitable regardless of its size (required for self-funded businesses) versus a business designed to grow as fast as possible regardless of profitability (required for "winning the market," or for significant investor returns, or for making an impact on a large number of people). Either is a rational goal, but each requires

Figure 2

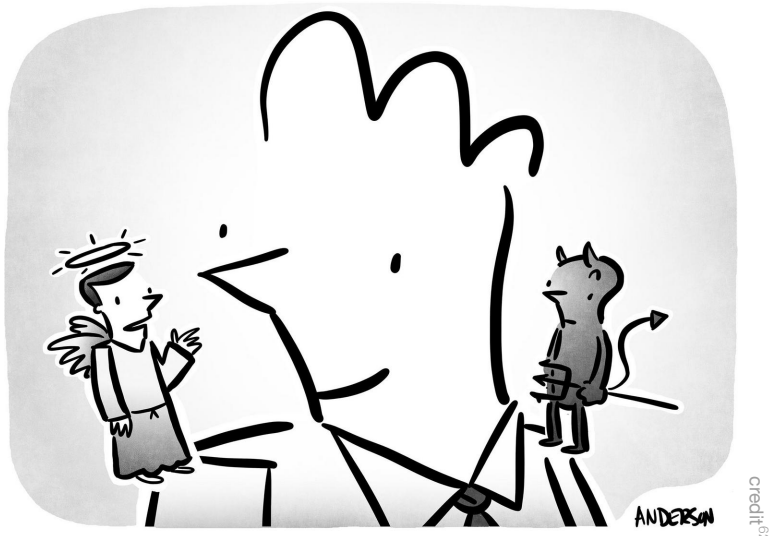
* WP Engine started with SMBs served by small agencies. Only after we reached $10M ARR, and had anchor enterprise customers, and enough funding to invest, and invested simultaneously in marketing, sales, support, and infrastructure for larger websites, did we decide to expand (p. 793) into larger customers.

different processes, mindsets, activities, and choices. Advice from the Internet often doesn't specify which goal it's designed for; perhaps even the advice-giver doesn't know (p. 751). Applying a mixture, or a "balance" of these ideas, reduces the chance of either outcome.

Another example is targeting niche versus broad markets. Niche companies need to speak only to that niche; broad must appeal to a wide variety of people. Niche needs to charge more, because there's fewer potential customers and because more value can be delivered per customer; broad needs to be inexpensive, as most people and businesses have little money, especially outside of the West. Niche is delivered in one language; broad thrives with forty languages.

This is also how to resolve the apparent paradox in all startup advice: For anyone who tells you "Successful companies do X," you can always find someone else saying "Success comes from avoiding X." Indeed, it's easy to find examples of both successes and failures, for each opposing admonishment! This is the "startup advice" equivalent of the old observation that "proverbs come in pairs." Example: "They who hesitate are lost" but also "Good things come to those who wait." Or "Many hands make light work" but also "Too many cooks spoil the broth." All are true—that's why we repeat the bromides—the solution is to pick the advice that best matches the context *you* are in, and the goals that *you* have.

Often people claim "it's a balance" to justify their refusal to make a decision. Whenever you're tempted to claim The Middle Way, pause to make sure that is the correct approach to solving the contradiction.



"You know what? Yes. Go ahead. Live a little!"

"IT'S A CHOICE… TO A LIMIT."

This modification to "It's a choice" is the right approach when one choice is the answer, but the other choice cannot be completely ignored.

The traditional VC-backed company is trying to grow as fast as possible, and thus traditionally it spends more money than it makes. We would say: It resolves the choice between maximizing growth and maximizing profit, by maximizing growth.

However, nowadays (2024) both private and public markets are rewarding companies only* if they have at least some profit. Valuations might be several times higher for otherwise-equivalent companies with –5% versus +5% profit margins. Therefore, the current

* Except for AI startups. (p. 419)

choice among these companies is to maximize growth, *but only if* profit doesn't dip below some threshold.

This does not mean it's a balance; it is not a middle way, not a compromise. The goal is still to maximize growth. It's that costs are a *constraint*. There are many examples where we have made a clear choice, but we're constrained by some limit.

In this mode, the answer is "Maximize A, while satisficing B (p. 887)." One is the goal, the other is the governor. We turn our attention to B when it's in trouble, otherwise we focus on A (Figure 3).

This solves the puzzle of whether we should sacrifice our health (p. 1547), our social relationships, and our physical or mental health, for the success of our startups. The fact is it takes an inordinate amount of time and effort to get a company started.* It does involve sacrifice (p. 901). However, there are a numerable cases where people sacrificed

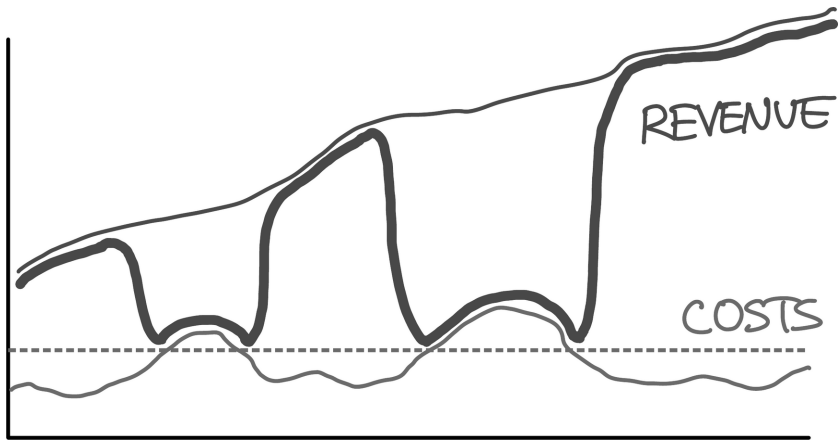


Figure 3: The dark line follows our attention over time: Mostly on revenue, but jumps over to costs when they grow beyond our pre-agreed threshold.

* The people who tell you otherwise have either not created the type of company you are creating, or they're far past that point, now with the luxury of "work/life balance," telling you that it was possible to live like that all along, despite they themselves not succeeding in that manner.

their physical health so much that they becoming seriously ill, or sacrificed their family so much that they earned a divorce, or sacrificed their mental health so much that they suffered a breakdown. In a few famous cases, it even led to suicide.

The answer is not a "balance," i.e. not working too much. The answer is that you have to work as hard as possible without falling off the cliff. The edge can be hard to see, but the answer isn't to back off so far as to never approach it; the answer is to toe the line, and back off slightly when you feel your feet slipping.

The best way to execute this, is to establish an objective measure of the thing that should be satisfied. That way, it's always clear to everyone whether it's currently satisfied (in which case we should spend no time or energy on it), or whether it's not (in which case we need to attack and fix it as quickly as possible). This is one of the critical use-cases for KPIs (p. 645) and is built into my method for quarterly planning (p. 1065).

"WHY NOT BOTH?"

This is the correct approach to create novel, insightful, unique products and strategies.

Before Tesla, there were electric cars with low performance, and gas-guzzlers who could go from a full stop to exceeding the highway speed limit in three seconds. There were car models which optimized for one extreme ("it's a choice") and hybrids with improved gas mileage but still low performance ("it's a balance").

Telsa created a third option: "Why not both."

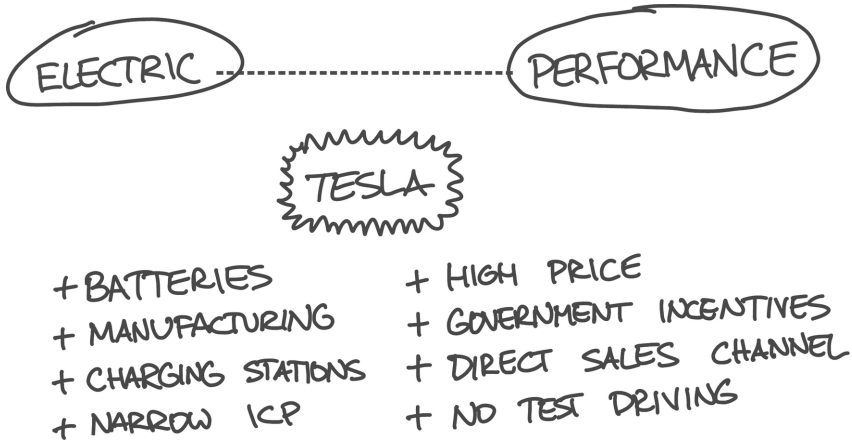
Before you get too excited, note this required several things:

1. **Invention.** Something new had to be created. In this case, new technology (batteries and car design) and new distribution channels (selling directly instead of through dealers).
2. **Risk.** There was a high likelihood that it would fail. The world is littered with failed "why not both" startups, whose post-mortems bemoan (p. 379) the "lack of focus" that came from "trying to do too many things at once" instead of "doing one thing and doing it well."
3. **A set of additional decisions.** It wasn't enough to just invent a new battery; a series of other decisions had to be made: placing batteries in the floor to maximize handling and safety. Creating new manufacturing facilities. Spending hundreds of millions of dollars on recharging stations to address range anxiety. Targeting a market segment that valued novelty over reliability, valued performance over family transportation, and could afford an extra luxury car. Multi-national lobbying for federal donations both to Tesla and in consumer incentives. Cutting out the usual sales channel of dealerships, replacing it with cult-of-founder marketing and PR, so that prices weren't astronomical, but which meant potential customers couldn't even test-drive the car.

Synthesizing a new solution, which allows us to select "both," but only if many other things are also true, is the most difficult thing to do, but also by far the most valuable.

Because it requires invention, others aren't doing it. (Competing car manufacturers didn't copy the batteries, even after Tesla open-sourced their patents.) Because it requires risk, others are unlikely to follow even after you start. (It took more than a decade for other car manufacturers to even attempt to follow Tesla, and even now Tesla is unrivaled in performance and safety.) Because it requires a network of other self-reinforcing decisions, it creates a unique offering, which remains unique even if competitors copy a subset of your strategy.

In short, it's a moat (p. 761)—you are unique and unassailable, even by well-funded competitors.



Often the press will focus on the "invention" aspect, and indeed that's the most interesting and press-worthy thing to talk about. But the real key is the set of "other things which also have to be true."* These additional decisions are what convert pure invention and wishful thinking into a complete strategy.

In particular, this means adding more constraints, more decisions, *even things which are undesirable for customers or for the company*, which together create an interlocking, self-reinforcing web. The undesirable outcomes are acceptable because there are always customers who value the positive more than they despise the negative (p. 317), and thus you are uniquely the best product for them.

So Zappos was able to offer free shipping and free returns in apparent conflict with also having low prices, by creating scale through fiercely loyal customers, which in turn was generated by having remarkably amazing customer service, which in turn was a result of a unique culture giving unprecedented autonomy to customer-facing

* A phrase made famous by Roger L. Martin,[623] who suggests that great strategies come from taking the several most critical things that we need to achieve in order to win, and then adding to that: "What else would have to be true?"

people. One negative side effect was low profit margins (so the resulting sale to Amazon at $1B was at only 1x revenue).

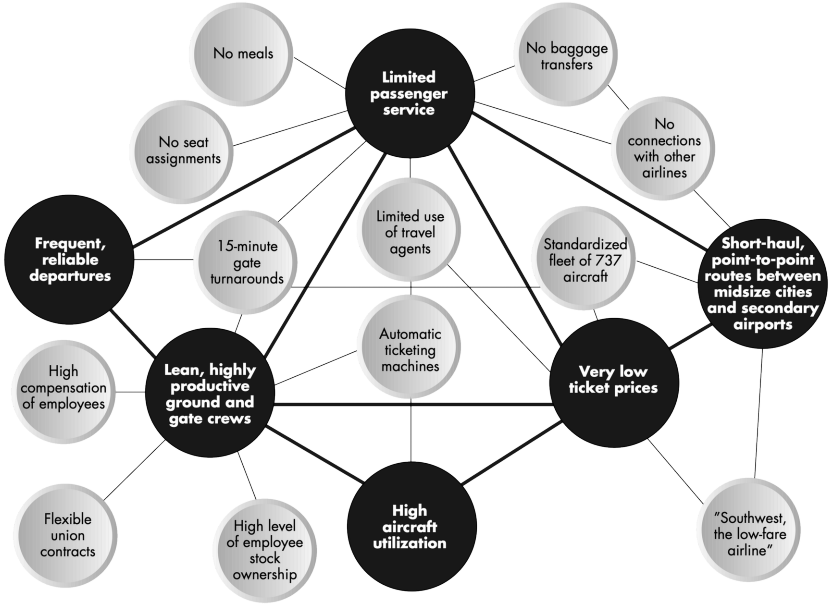
Or Southwest Airlines was able to offer frequent departures on planes that weren't completely full, yet with the lowest fares in the industry, due to an exceptionally low cost structure. They had low costs because they used only one type of plane, ran only a few, short routes, and earned extreme loyalty from employees through a nurturing culture. The negatives included only having a few routes, and short ones, no international, no food, and more. This is more fully documented here (p. 543); for decades they've remained true to their network of decisions (Figure 4), and have the highest profits of any airline in American history, and the only one to never have gone bankrupt.

This is a common refrain in strategic frameworks. For example, the central idea of *Blue Ocean Strategy*[625] is to create a new category of product through a combination of choices that are "contradictory" according to existing categories, while also solving for profitability. Seemingly impossible, certainly difficult, but nevertheless a powerful result when it is achieved.

The courage to make all of these decisions together, to accept the unique advantages as well as the undesirable consequences, is what creates the best business strategies in the world, and is the complete way of saying "why not both."

Don't be so quick to say "it's a balance." You might be avoiding a hard decision.

Or better yet, turn an apparent conflict into a unique strategy that others cannot copy.

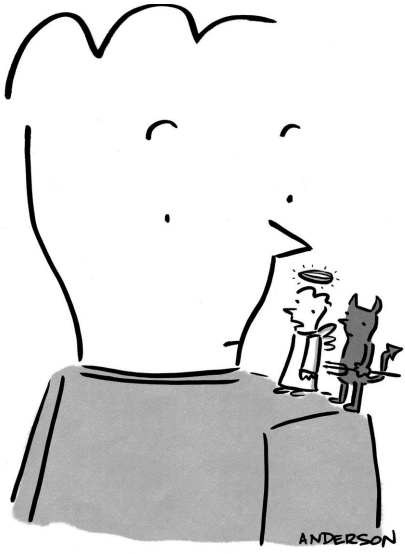


credit624

Figure 4: Michael Porter's "Activity Systems Map" for Southwest Airlines, from his 1996 article "What Is Strategy?" Harvard Business Review 74 (6): 61–78.

Chapter 38:
Binstack: Making a maximal multi-dimensional decision

RUBRICS · BINSTACK · REAL-LIFE ERRATA



"Do it, don't do it. We don't care anymore."

credit 626

Many decisions in life and business are instances of "multi-dimensional maximization," in which we wish to pick the "single best" among a set of choices, but we're confounded because each choice is variously better or worse along different dimensions.

Examples:

- Which major feature should we spend the next six months building?
  (P would generate revenue, but Q would reduce cancellations, but R would save us money)
- Which candidate should we hire?
  (P has the best skills, but Q has more experience in our market, but R seems like the best culture-fit)
- Which new marketing campaign should we spend thousands of dollars to test?
  (P is cheaper to try, but Q has a larger reach, but R is targeted at our industry)

Not only do you need the best decision, you also need to be able to **explain your decision to others**, especially to those who wish the decision had gone a different way. Do not under-value the importance of crisp explanation.

The "rubric" is the typical framework for these decisions; a separate paper on this site explains how to use one effectively for "ROI-style" decisions (p. 171).* Unfortunately, while it may feel productive to fill many cells with many numbers, and while it may feel analytically rigorous to convolute those numbers into a final score, this fails to clearly identify the best choice, and fails to create a clear explanation for the choice.

---

* The goal of ROI is to maximize efficiency, i.e. deliver the most amount of value per unit of time. This paper asks a different question: How to decide the single most valuable thing, regardless of cost, with incommensurate and conflicting dimensions (p. 589)

After demonstrating and analyzing the causes of this failure, I present an alternative framework I've nicknamed "Binstack."*

WHY RUBRICS DON'T ADD UP

Consider two players in a game, with attributes:

Attribute	Player P	Player Q
Health	8	5
Strength	5	7
Speed	2	6
Endurance	5	2

Which player is better? If one scored higher than the other in every dimension, the choice would be simple. In this case, each player is better than the other along two dimensions, and worse along two; objectively there's no clear winner.

So let's try a rubric. In its simplest form, we add up the scores contributed by each dimension, and the total score decides the winner. Unfortunately, this doesn't tell us which one is better:

* Additional resource: Adam Waselnuk[627] created a Notion Process Template for Binstack.[628]

Attribute	Player P	Player Q
Health	8	5
Strength	5	7
Speed	2	6
Endurance	5	2
Total:	**20**	**20**

Games often engineer this result on purpose, to create players that are different but not over-powered. This creates a balanced game, but to make a confident decision about which one is "best," we need something imbalanced.

Real-life rubrics often look like this example: a pile of options that share a similar score, resulting in no clear winner. Even if we make a decision, we can't explain the decision to others, because in actual fact it's a tie, and the tie was broken arbitrarily. That's no way to make a decision.

To create separation, people often add "weights" to the raw value to create a new kind of "score." In this case, suppose that with our game-playing style we don't care much about Speed, but we can really leverage Strength. So we assign weights which we multiply against the scores, to compute a customized metric of "value."

Attribute	Orig P	Orig Q	Wt	Wted P	Wted Q
Health	8	5	1.0	8	5
Strength	5	7	2.0	10	14
Speed	2	6	0.5	1	3
Endurance	5	2	1.0	5	2
Total:	20	20		**24**	**24**

That didn't help. What if we contrive to *force* Q to be better, by intentionally using weights that penalize the two attributes where P is superior?

Attribute	Orig P	Orig Q	Wt	Wted P	Wted Q
Health	8	5	0.5	4	2.5
Strength	5	7	2.0	10	14
Speed	2	6	0.5	1	3
Endurance	5	2	0.5	2.5	1
Total:	20	20		**17.5**	**20.5**

Even with a conspiracy to throw the election for Q, it wins by a mere 2.5 points out of 20—hardly a resounding victory that would give everyone confidence in the decision.

This happens in the real world. Even with weights, a clear winner often does not emerge, and again we're back to a weak, indefensible decision.

Worse: In the real world we rarely have precise scores. Attributes like "potential new revenue" and "increased customer satisfaction" are not predictable (p. 193) with accuracy. For more qualitative measures we use scales "from 1 to 5" which are even less precise. This imprecision creates inherent error, which is then compounded by multiplying weights. Differences in the final results are more error than signal.

Worse again: The attributes aren't comparable. Whatever the unit of measure for "Health," it isn't related to the unit of measure for "Speed." By adding them together, we're implicitly saying "these are comparable," but they are not. Weights are supposed to solve this by converting everything into some sort of "value," but if you say out loud what the weights are doing, it sounds incorrect. For example: "Every $150k of additional ARR is exactly as valuable to us as 10% more customer satisfaction. Indeed, we would be fine with customer satisfaction going down 10% if we added $300k in new ARR." Really?

It's largely noise, which is why we're unhappy with the so-called "winner."

SOLUTION: BINSTACK: STACK-RANKED BINARY ATTRIBUTES

To transcend the noise, stop pretending that the values in the rows are precise or comparable.

A general rule of complex decisions, is that often they're difficult because we've avoided making other, smaller decisions. That is the case with the rubric, as we will now fix.

Our top-level purpose is to pick the item that maximizes impact, and to be able to explain our decision. First we're going to make smaller decisions about the true impact of each item, and then we're going to make a decision about which impacts are most important.

Binary materiality

No more values, no more weights, no more scores. Either an item *materially* contributes to that attribute, or it doesn't. "Materially" means the effect is so large you can measure it easily:

- Not just "more revenue," but at least a 10% bump so that the curve visibly changes.
- Not just "more retention," but a 20% decrease in cancellations tied to a specific cause.
- Not just "more intuitive," but a 40% decrease in support tickets on a certain topic.

- Not just "more competitive," but sales will add it to their standard presentation and marketing will add it to the feature-table on the pricing page.
- Not just "more profitable," but overall gross profit margin will improve by 1%.
- Not just "will pay for it," but putting it in a higher pricing tier or add-on will cause 5% of customers to upgrade.
- Not just "better UX," but a 50% increase the success-rate for people moving through the interface.
- Not just "widely used," but 40% of customers surveyed scored at least 4 out of 5 on whether they'd use the feature.
- Not just "customer satisfaction," but moving from a 4/10 to a 8/10 on a survey related to this area of the product.
- Not just "thought leadership," but marketing commits to getting ten external articles to reference it in the next quarter.

Force people to write down exactly what the material change is expected to be. Not because the estimates are accurate, but because it forces the person to think through the answer. Most ideas, we'll eventually admit, are so incremental that we won't be able to measure the effect; that means it is "not material." That's a tough fact to face (p. 657), but remember the point of the exercise is to force exactly these conclusions, to drastically reduce the field of ideas so that only the actually-best ideas remain. These smaller decisions make the larger decision tractable.

Because this "material change" is just a guess, we won't put it in an equation—no computing with noise! But if you can't justify a magnitude greater than "business as usual," the idea is simply not impactful enough. Your standards are higher than that.

In our example, if we simplistically considered any score that is "6 or greater" to be "material," we'd already have a winner:

Attribute	Player P	Player Q
Health	✓	
Strength		✓
Speed		✓
Endurance		
Total:	**1**	**2**

Each "point" in this method is meaningful, so even a difference of 1 point crowns a clear winner. With real-world attributes, and a sufficiently high bar, you will reject nearly all items quickly. People won't like that—their favorite thing will be cut—but it's the only way to stop wasting your time dithering between a pile of things that won't make a difference.

It's also extremely easy to explain your decision: Q materially impacts two important things; everything else is less impactful.

It is, of course, still common to have ties. Indeed, if in our example we considered anything "5 or greater" to be material, it's back to a tie:*

Attribute	Player P	Player Q
Health	✓	✓
Strength	✓	✓
Speed		✓
Endurance	✓	
Total:	**3**	**3**

To address this, we need one more rule.

* Although in this case perhaps the problem is that our standards for "materiality" are too low, as opposed to the options being too good.

"I haven't yet decided if I'm staunchly in favor, or staunchly opposed. But rest assured, however I vote, it will be staunchly."

credit[629]

Stack-ranked attributes

Back to the top: (1) We're trying to isolate the *one* thing that would be *most* impactful, and (2) complex decisions feel impossible because of a lack of smaller decisions. We've made some decisions already, but we need a few more to isolate the one winner.

We have to decide which attributes are most important. Currently we are treating all attributes as equally important—a check mark next to "Endurance" is equal to a check mark next to "Health," but is that really true?

With a standard rubric, the fact that all attributes are not equally important drove us to reach for "weights." But that computation confounded us with noise. Instead, we simply order attributes by importance, in a single ranked list. No numbers.

"Simply order them" is easy to say but not easy to accomplish, because people get stuck in circular debates:

> "Growth is more important than profit, because it's possible to optimize our costs later."
>
> "Yeah, but if we're unprofitable on a unit basis we'll cause a cash-crunch, so we have to be profitable first."
>
> "Yeah, but if it's only about profit, the best thing to do is just 10x prices, and whichever customers stay are super profitable, but that would be wrong."
>
> "Yeah, but if it's only about revenue, the best thing to do is to sell $1 bills for $0.80, and that would be wrong."

This conflict highlights the "smaller decisions" that still need to be made.

Both sides are correct in saying it's bad to maximize one thing with no regard to any consequences. But surely you've already ensured your list contains nothing outright absurd.* So these *reductio ad absurdum* arguments are moot and can be ignored. Assume (and enforce that) the ideas are sensible, then decide what outcome is most important.

A typical ordering for a VC-backed B2B company, optimizing for "growth is paramount, because if growth is there, we'll be able to raise more money," could be:

1. Revenue growth (i.e. "the single biggest driver of equity-value")
2. Number-of-customers growth (i.e. "market share")
3. Product experience (i.e. "customers love of the product")
4. Support cost (i.e. "a cost; more importantly, a measure of usability")
5. Infrastructure cost (i.e. "a cost; should organically improve with scale")
6. Net-profit expansion (i.e. "profitable business model")

---

* If you have trouble ensuring that items are meeting basic standards, create criteria for an idea making it onto the list to begin with. Examples: it can't be less unit-profitable than some pre-determined target, it can't take longer than N sprints to execute, and it can't require significant retraining of the support team.

A bootstrapped company designed to create wealth for its founders and employees, while being a place where employees genuinely love coming to work and customers genuinely love the product, might be:

1. Cash-basis profit expansion (i.e. "wealth creation" + "mandatory to keep the business alive")
2. Product experience (i.e. "the reason we get up in the morning is building a great product people love")
3. Fun (i.e. "I built this business to be the place I *want* to work for")
4. Number-of-customers growth (i.e. "stagnation is the prelude to death")
5. Minimizing number of employees (i.e. "we joined a small company to avoid bureaucracy")

These are generic attributes; "more revenue" could include almost anything. It's better if your lists are more specific, based on current circumstances, or focused on a subset of the strategy. For example, suppose a product that targets mid-sized restaurant chains is having trouble with customer retention. A better list might be:

1. Dramatic increase in usability
   The #1 reason customers give us when they cancel in their first year, is that training their employees is too difficult, so the software never gets used.
2. Reduce costs
   Price is the #1 reason that long-tenured customers give when they switch to a competitor; reducing costs means we can reduce prices while generating the same profit.
3. Increase market-differentiation
   If there were features that customers couldn't get anywhere else, they would stay despite (1) and (2).

Note how obvious corporate goals like "grow revenue" and "happy customers" are embedded in these goals, but insights or data produce

more specific immediate goals. This will cause even better ideas to be selected, and will help the team brainstorm better ideas in the first place.

Binstack: The final process

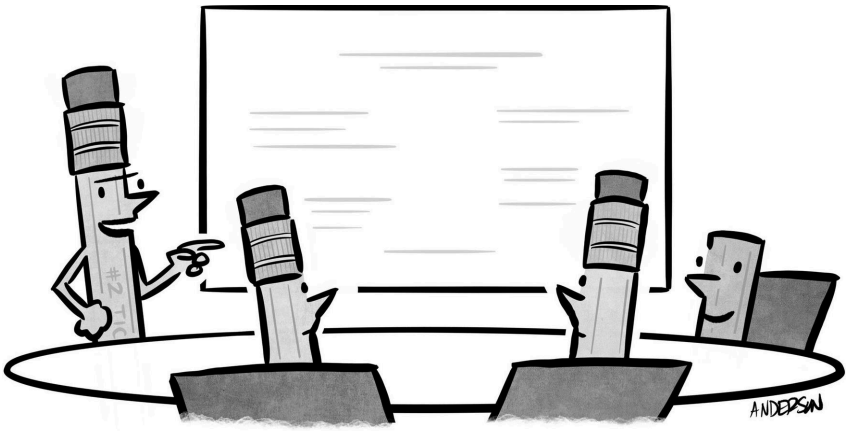
You've finally made all the "small decisions" that make the big decision clear. With your stack-ranked attributes and binary scores of which items materially affect which attributes, here's what you do:

1. Cross out items that don't materially address the top-ranked attribute.
2. For the second attribute...
   a. If no tasks address it, move on to the next attribute.
   b. If exactly one remaining task addresses it, that's the winner; you're finished.
   c. If multiple remaining tasks affect it, cross out all the others and continue on to the next attribute.
3. Repeat step (2) for the third attribute, fourth, etc.

These steps honor our smaller decisions about which results are most important to manifest (ordered attributes), and what these options *really* accomplish for us (binary materiality). It also ensures that we'll materially affect our #1 attribute; even if another idea moves several other needles, we still have to honor our "small decision" about what is of paramount importance.

The final decision is trivial to explain. It goes something like this:

> We decided the most important things we have to accomplish in the next few months are to grow top-line revenue and create defensible technology. Item P does both of these things; none of our other choices did.



credit 630

"I like you, Anderson! You're decisive!"

Or defending why you didn't pick some other item:

> Item Q is a solid idea; indeed it would both increase profitability and increase our differentiation in the market. However, the most important thing right now is to grow revenue, and item P accomplishes that whereas item Q doesn't. However, in future, if we change our priorities, or complete item P, item Q will be wonderful to consider!

REAL-LIFE ERRATA

"Effort" conspicuously absent

Many rubrics are set up as "ROI" calculators, i.e. measuring impact relative to the cost of achieving that impact. This often results in selecting less-impactful items, because they are cost-effective to implement.

Sometimes that's the right choice, but Binstack is about selecting only for maximum impact, not cost-efficiency.

If you actually want to maximize ROI, use this method (p. 171). And here's how you decide (p. 221) which method is right for your current decision.

Fun is underrated. Add it in.

It's still possible to tie. If you have lots of items left over, perhaps your materiality threshold isn't high enough; raise it to thin the herd.

Supposing you have two ideas that are truly indistinguishable, you could flip a coin. I don't recommend that, because you can't explain your choice. A person passionate about the choice you rejected would be upset to hear you ruled against them so flippantly. Instead, pick whichever item the folks doing the work *want* to do. Do what's fun.

It still sounds flippant. What business does "fun" have in business? When people work on something fun, they work harder and better while enjoying themselves more—more productivity yet more happiness. Do not dismiss this life-hack.

Of course we cannot do what's fun at the expense of what *needs* to be done, but when those two things are not in conflict, why would you not round off in favor of fun?

If you like this idea, take it further: Put "fun" in the attribute stack-rank, and rank it high. Even second position is not ridiculous. Knock that #1 priority out of the park while having fun. What's wrong with that?

Evolving stack-rankings, and different stack-rankings per team

You should expect the stack-ranking to change over time, even rapidly. In the early days of a company, just getting any customers is hard,

at any price, so that might be much more important than revenue or profit. A mature company who reliably gets customers in the door might be more interested in expanding efficiency or profitability, not because growth is unimportant, but because it's so systematic that it is no longer an existential threat, and other things are more pressing.

If you have multiple teams, and therefore the time to execute multiple items, you might want separate lists for different goals. For example, you might say, "We want one initiative that will materially increase profit, one that materially increases our internal effectiveness or efficiency, and all the rest should maximize growth." Each of those would be the number-one item in its own stack-ranked list, and the attributes below that might be copied from the company-wide general list.

In all cases, remember to hone the attributes with more specificity, to generate better ideas.

> *"I've generally found that the best product ideas live at the intersection of "duh" and "holy shit."*
>
> *–Aaron Levie*[631]

What if nothing is left?

Sometimes we're so harsh with our materiality threshold that none of our ideas meet our exacting standards. What does that mean?

It means your ideas aren't good enough. It means your problem wasn't one of prioritization after all, but rather of not having ideas worth prioritizing.

Focus the team on this new, more pressing problem: To generate better ideas. Here's some help with generating better ideas (p. 53).

But this will take time! Fine, put it in the sprint. But we need a plan right now! Too bad; it's better to take a month to find a wonderful thing to spend the rest of the year on, than to plod along doing things that aren't valuable enough. Do high-ROI (p. 171) small projects in the meantime.

Why bother scoring everything when most will be rejected immediately?

Indeed, you needn't bother. That saves time.

However, remember that an important aspect of decision-making is explaining your decision. Often, explaining "why not Q" is just as important as explaining "why P." Other people will want to know that you seriously considered other options.

Most importantly, there are people who really wish you had selected Q. Maybe Q was even their idea. The rejection will be easier to accept if Q was seriously and genuinely considered. Perhaps, by being invited into the decision process, the person who invented Q will come to the right conclusion on their own. This is important; do not underestimate the human—and humane—part of the process.

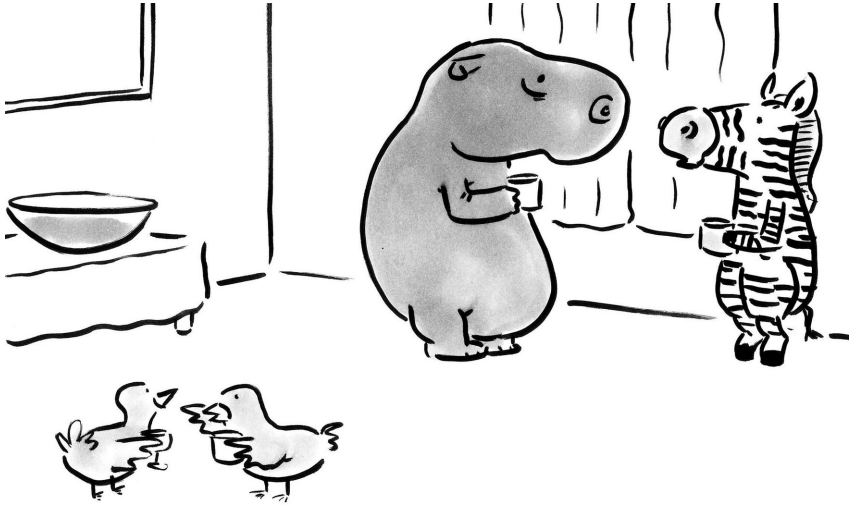
Reductive

Binstack can feel reductive—over-simplified, ignoring the reality of a complex world, therefore resulting in an incorrect conclusion. It is true that for complex problems like foreign policy, national economics, and climate change, a reductive approach is invalid.

But for finding the right feature to build, or the right marketing campaign to launch, or the right bug-tracking software to adopt, or the right database to use for a new project, or the best candidate to hire, being (intelligently) reductive is how you transcend the noise, arrive at a clear decision, and explain it to others.

This isn't foreign policy, it's a feature list. Make an impact with Binstack, and be happy!

Chapter 39:
Never say “no,” but rarely say “yes.”



“I know he’s not much to look at, but the symbiosis is great!”

Everyone says small startups require focus. Say "no" to anything that distracts from your goal, your vision, your strategy, tempting though it is to explore all opportunities, hoping each time that *this* is the one that will catapult you to "success (p. 861)" (whatever that is).

Lack of focus results in half-assed initiatives, each interrupted by apparently greener pastures before you've invested the time and devotion it deserves. Learn to say "no!"

Ah, but then again you must also experiment with new ideas. Fail fast (p. 1261)! Pivot (p. 193)! Test (p. 239)! Doubt (p. 457)! Always be collecting evidence that you're wrong, always be trying new things in case you've been blind. Never pass up an opportunity to change, learn, grow.

So... how are you supposed to explore other ideas if you're also supposed to be saying "no" to anything that diverges from The Plan?

Here's what I do: I never say "no." But I carefully qualify "yes."

I learned this trick in high school. In the mid '90s it was clear that Apple had lost the personal computer battle and all their developers were fleeing like rats off a sinking ship into the ocean of opportunity that was Windows 95. As a maven of the Macintosh API and still willing to admit it, I landed small contracting jobs fixing up code that other developers wouldn't touch.

My typical rate was $25 per hour, which feels like a lot of money when you're 17 in the '90s.

One day I got a call from some poor schleps I didn't want to help. They had just completed a new product written in Java and it was broken on a Mac, and could I help? None of their customers used Macs, so they didn't think Macs were important, but then it turns out the main investor is keen on seeing the demo on a Mac, and when they tried it, it didn't work. (Yay investors! Yay supposedly-cross-platform-platforms-that-aren't-really!)

I wanted no part of this. Java was brand new and known to be full of bugs, and anyway I was a C/C++ kind of guy, and I didn't want to get involved in an academic fad language like Java.*

I could have said "no." Given my specialty and my goals, traditional career (or startup) theory says I should have said "no."

But instead, on the advice of an older, wiser friend, I showed up at their office and said I'd do it for $100 per hour.

I fully expected them to laugh in my face. Maybe I would receive a condescending talking-to about the audacity—nay, the *impudence!*—of someone of my tender age and meager experience walking in here and demanding such outrageous compensation, someone who, let's be clear, is technically too young to even enter into a legal consulting agreement in the first place.

And then I would have slinked out of there, embarrassed but ultimately no worse for wear.

But that's not what happened. They looked me up and down, their faces painted with both incredulity and surrender. They said OK. An hour and a half later, everything was working. The difference between saying "no" and getting $150 for about two hours of my life was all in how I phrased "yes."

And I have stories that went the other way, which are just as important.

At WP Engine,[633] for example, we're constantly talking to large bloggers who want to move to our system. These are folks with big requirements—tens of millions of monthly page-views, traffic spikes, custom code, perfect up-time, and 24/7 support.

Should we take on those clients? Maybe not—after all the stated goal of WP Engine is to serve the "middle market"—the folks who have outgrown free blogs, don't like maintaining their own servers, but aren't so large that they have extreme hosting demands. That's our profitable niche.**

---

* So yeah, I simultaneously decided that (1) Java is a fad and (2) I'm sticking with the Macintosh Toolkit; five years later one those platforms had zero developers and the other had one million, and I picked exactly wrong. Predicting the future (p. 193) is hard.

** Editor's note: This was written about 1 year into WP Engine's life; in 2023 WP Engine is 13 years old with 200,000 customers, and powers more large

Or so we think. But if we just say "no" to these big bloggers, maybe we're closing the door on big, important orders. Perhaps the entire company should pivot—maybe it's easier or more profitable to serve 100 large blogs than 1000 medium ones. But how do we know if we say "no?"

Then again, if we say "yes" we might really be screwed. If we can't provide them the human and technical service they expect, now we've hurt a blogger, we get bad press, and we've wasted a bunch of time. Or even worse, we hold on for dear life but it's extremely unprofitable, and now we have this expensive, time-consuming albatross around our necks.

So we've said "yes" by quoting high enough that we know for certain we will make good money on the deal, so much so that it will partially fund something else we want to do. Maybe that means a big new advertising campaign, or hiring another WordPress expert for our staff.

There was one especially large customer where we literally thought of it like this: This deal needs to be big enough to not only make a reasonable profit on the operating expenses, but pay for an entire developer's salary (assuming bootstrapped, put-in-elbow-grease-for-stock low salary), because we know this new customer will occupy a lot of that person's time, but all the remaining time we get "for free."

So we've given a lot of qualified yeses, and many were rejected. (P.S. Now we're able to say "yes" to those same bloggers, but that's because over time we've taken on bigger and bigger customers, and now a blogger with 30 million month page-views is something we know we can handle.)

At Smart Bear[634] I used this principle yet again. Companies would fly me out to help them implement a peer code review process, which half the time actually meant that "management" wanted me to con-

---

WordPress sites than anyone on Earth. This *was* a great segment to start off with, but we evolved as we scaled.

vince everyone else that code review was a good idea, and invent a process painless enough that they might actually do it.

From a business perspective, this was a poor use of my time. These folks had already bought our software, so it didn't sell more seats. When you counted a travel day on either side of the engagement, the time I lost could have been spent landing just one additional customer or make some important changes to the code, either of which almost certainly makes us more money.

Therefore, initially I just said "no." But again that's wrong. Eventually I said "yes," but the price was $2500/day *including travel days*, which for these sorts of engagements is unheard of. (Typically you get reimbursed for travel expenses but not paid for that time.)

This immediately cut out most trips, but some remained. On those trips I'd haul in $10,000 for a week of easy work, which I'd often combine with a long weekend with my wife. And anyway those people *really wanted me there*, which made the work that much more enjoyable.

So the principle is straightforward: Set the conditions of "yes" such that:

1. **If they say "yes," you're happy** because the terms or money are *so good*, it more than compensates for the distraction, perhaps funding the thing you really want to do.
2. **If they say "no," you're happy** because it wasn't a great fit anyway; it's not a worthwhile return on your time and effort.

So that's the punch-line, but before you leave I'd like to **overemphasize the idea of "funding the thing you really want to do."**

This can take many forms, but it's the single best way of figuring out *how* to qualify your "yes." Examples:

- "Yes" if it pays for an entire additional person.
- "Yes" if this extends the runway of our startup by at least three months.

- "Yes" if it completely funds development we'd like to do anyway.
- "Yes" if it means one of the co-founders can quit her day-job.
- "Yes" if it will completely pay for three new marketing efforts.

Think of it like another form of funding. **Funding is always a distraction from actually running your business, so the amount of money you get must be transformative to the business.** Each of those bullet points *are* transformative, in that each has the potential to move your company from "hobby" to "real business."

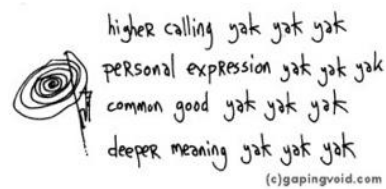
And if they say "no," you're fine with that, because it would have been a distraction which wouldn't have moved the needle.

Chapter 40:
"Authentic" is dead. And so is "is dead."

IS DEAD · WHAT TO DO

It's time to retire the following phrases. They should no longer be used in any context except derisive mocking:

- The Holy Grail of
- Designed with you in mind
- Putting customers first
- All new
- Win-win
- Proven track record
- Fast and easy (p. 1589)
- The leading provider of[636]
- And more[637]
- For everyone (p. 317)
- The go-to choice for _____
- Transforming the way you _____
- Designed with ❤️ in _____

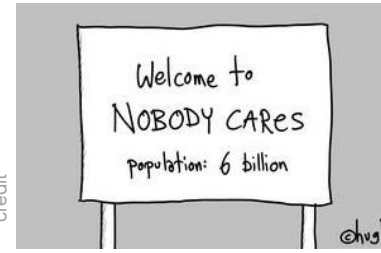


credit[635]

- The future of _____ is here
- Everything you think you know about _____ is wrong
- _____ considered harmful
- _____ is broken
- _____ is dead
- _____ sucks[638]
- Saves you time, so you can get back to doing what you do best.

Also eschew these words, as devoid of meaning as a yogi's mantra and as useless as a simile that doesn't contribute new information:

- **Authentic**
- Solution
- Turnkey solution
- Genuine
- Powerful
- Secure
- Simple
- Innovative
- Insight
- Revolutionary
- Unsurpassed
- Unparalleled
- Cutting-edge
- World-class
- State-of-the-art
- Game-changing
- Ground-breaking
- Value-add
- Disruptive (p. 1509)



credit[639]

These words have been corrupted by those who claim to honor their meaning but do not act accordingly.

When a company claims to "put customers first" but then uses "Level 1 support" as a shield[640] to prevent customers from intruding on profits, we realize talk is cheap.

When a company claims to have "secure" payments but then 100,000 credit card numbers are stolen, we realize you don't need a permit to claim that you're secure.

When a company claims to be "innovative and disruptive (p. 1509)" but then pitches an idea you've heard ten times in the past month, it reminds us that if you have to say it, it's probably not true.[641]

When 78% of "About Us" web pages claim "the leading provider" of something, we are no longer impressed.

Like a song over-played on the radio, like a restaurant over-hyped in the magazines, **repetition of even powerful, wonderful phrases will kill them.**

Oh I know 21% of you stopped reading as soon as you saw that "authentic" made the list, and shot over to Twitter to unleash a scathing missive explaining how "authenticity" is the prime mover[642] of modern marketing, honorable salesmanship (p. 751), and meaningful relationships.

I agree! In fact all these words and phrases *theoretically* carry meaning, but theory is for people who don't need to sell $2,600 more software by next Friday so they can make rent.

And sure, it actually *is* good to be "authentic." I respect the work of all those bloggers and Twitter-ers and lecturers and consultants who drove this word deep into our psyches. Indeed, it is a gift: bringing genuine authenticity and the give-first-sell-later behavior, in contrast with to traditionally sterile, aggressive, non-engaging, selfish world of sales and marketing. The more people honor this code of conduct, the better for us all.

Nevertheless, it's time to retire words like "authentic." The misuse is to too widespread, the abuse too deep.

credit[643]

"Buzzword jargon buzzword, hyperbole buzzword buzzword, trite rhyming platitude… Yep, looks good."

WHAT SHOULD YOU DO INSTEAD?

Be specific.

Many of the dead words weren't especially illustrative to begin with. As far as I know, a "solution" just means "product and/or service," so the word isn't adding information. Instead, inspire me by being specific.

- Instead of "easy" say "so straightforward, you won't need a manual."
- Instead of "inexpensive" say "just a dollar a day."
- Instead of "powerful" say "processes 6,253,427 requests daily."
- Instead of "secure" say "blocks 96 million attacks daily."
- Instead of "disruptive" say "72% of our customers say they'll never go back to a normal email client."

- Instead of "beloved" show your NPS or CSAT or G2 rating or a massive wall of real, unsolicited testimonials, like tweets.

Here's more about being specific (p. 1439).

Show, don't tell

Some dead words are descriptive, but they don't paint a picture. "Powerful" sounds nice I suppose, but how does that change my life? *Showing* something in action is more evocative than *describing* it.

- Instead of saying it's fast, show a speed test (especially against competitors).
- Instead of saying it's easy, have a video demonstrating your tool solving someone's problem in 60 seconds flat.
- Instead of saying you have eager, responsive, intelligent tech support, put a "chat now" bar on every page of your website.
- Instead of saying that customers love you, show testimonials from 100 customers. (No one will read them all, but you can impress with sheer quantity.)
- Instead of saying "we're innovative," show your change-log, impressing the viewer with your product velocity.
- Instead of a bullet-list of benefits, quote actual customers describing your impact on their lives.
- Instead of generic-sounding testimonials, reproduce unsolicited tweets that show genuine love and gratitude.
- Instead of saying you value your customers,[644] tell them to call you to test it out, and then answer the phone on the second ring.

Name & Embrace

My favorite way to start a sales pitch is to make fun of typical sales pitches. What I always said at Smart Bear:

> I know you were hoping for a 22-slide PowerPoint deck with our mission statement and company history. I'm really sorry to disappoint! 'Cause I'm just going to start the demo and let you interrupt me with questions.

And:

> People claim that peer code review tools will do magic things like make your developers smarter or fix existing social problems with the team. Actually, if anything, code review can magnify latent social issues! However, I do believe our tool will save you time and aggravation in these 4 specific ways [...], so as we go through the demo, see if you agree.

Because you're willing to say the quiet bit out loud, to say what others hide, especially when we all know the truth, you earn credibility. (And often some laughs.) Now folks are more open to your claims—even those that are well-worn.

Own it completely

You *can* still use an abused word if you totally, 100% own the concept.

You can claim "legendary customer service" if you back that with first-ring, human phone service, online chat from your home page, quick-response Twitter monitoring, and 15-minute turn-around time on tech support emails even at 3am on a Sunday. Be sure to communicate all that too, because if you *lead* with the dead phrase I'll leave before you get the chance to prove it.

> *"Be the change you wish to see in the world."*
> *—Gandhi*

When old ideas become cliché, that's an implicit call for new ideas. This time around, can you lead instead of follow?

Of course this is unfairly difficult. Quick: Come up with a compelling new philosophy for human interaction and global communication, marketing, sales, and relationships!

Yeah it's unreasonable, and not certainly required, but remember the best ideas often aren't reasonable (excuse the clichés) groundbreaking, innovative, out-of-the-box, Earth-shattering epiphanies. **Often great ideas are a synthesis of other ideas** with just a smidge of novel insight, or just **putting into words what others sense but cannot articulate**.

This is the hardest and most time-consuming way to break out of the mundane, but also the most rewarding. And if you do come up with something, then your who will love you for it (p. 1559), and then help you grow and thrive (p. 259).

You're a little company, so act like one.[645] And enjoy the love.

Chapter 41:
The Code is your Enemy

BUILDER PROBLEMS · CODING LEADS TO FAILURE
PROOF FROM YOUR PEERS

"Ugh! I could feel them monetizing me with their eyes!"

credit[646]

YOU'RE A BUILDER. THAT'S GREAT. IT'S ALSO THE PROBLEM.

You're a builder, a creator—whether a back-end programmer, a Linux hacker, a Javascript ninja, a UX magician, a designer. You make stuff.

That's great of course, because in a new startup everyone needs to be either making stuff or selling stuff—there's no room for managers and executives and strategists. But **this also produces a natural weakness**, and when I look at what made me a successful entrepreneur—not just a great coder—one reason is that I discovered and overcame this weakness.

The weakness is the same as your strength as they often are (p. 569): Your love of creation. You love to write clean, tested, scalable, extensible, beautiful code. You love converting "JTBDs" into 960-wide artwork. You love developing an entire app in the browser against a scalable back-end.

And because you love it, you do it. You wake up in the morning thinking about what you can make, not how you can sell. You open Visual Studio before you consult your to-do list because there's something you just *need* to tweak. You launch xterm before your CRM (if you even have one, which you don't) because the server was running just a tad slower than you'd expect and you want to paw through log files.

The trouble is, this is almost certainly not the activity that would most benefit your startup.

As much as you're a minor deity when it comes to vim or Balsamiq, so are almost all the technical founders of all high-tech startups. We can all write code—at least well enough to get a product launched and

through a few iterations. We can all make a functional website—at least well enough to produce orders.

MOST STARTUPS FAIL, DESPITE EXCELLENT CODING AND/OR DESIGN SKILLS.

Read that headline again; this should bother you (p. 379) more than it does. Yet, because coding is your love, your passion, you just keep coding your way to failure.

Most startups fail because not enough people show up on the home page, or people show up but they don't try the product, or they don't pay for the product, or it's too expensive to get them to show up, or they don't tell their friends to come along, or because it's not solving a pain that people have, or it's not solving a pain that people know they have, or it's too hard to explain the pain, or a bunch of other things **that are *not* whether the code works or whether it looks good.** It's a Drake Equation (p. 667).

Customers don't open their wallets based on your unit test coverage or whether you used Bodoni instead of Times New Roman on the home page. In fact I've made millions of dollars on companies with hideously ugly websites (p. 853) and buggy code.[647] Those things are actually not the most important things (p. 1271).

So if these things—the raw materials and skills used in web-based startups—are necessary but insufficient, what are those things outside your comfort zone which nevertheless are the things that are actually valuable to your company?

Here are two:

1. Have you talked to 50 potential customers? By that I mean *fifty*, not a dozen. I once vetted an idea and after the first 10 interviews I thought I was really on to something. Suddenly things changed and future interviews weren't so clear. Turns out there was accidental bias in the people I selected, obvious only in hindsight. Another time, everyone said it was a great idea, but it wasn't (p. 845), which was only clear after dozens of interviews. Do you find it hard to locate that many people? Well it will still be hard to locate them after you've built a product, but then it's unlikely you built the right product, so now you've wasted months of time. So solve that hard problem now. Don't forget to vet the price at the same time (p. 515) and make sure they're actually going to buy it (p. 1469), otherwise it doesn't count. Here's how to interview. (p. 239)
2. Are people coming to your website every day? If not, solving that is *much* harder and *much* more outside your control than building software. Consider: Would you rather get hired as the CTO of a company with 1,000 daily new, unique, qualified visitors with a buggy product, or the CTO of a company with a clean, stable product and 10 uniques visits to the home page? You *know* you can fix the first case; no one knows if the second will ever be a real business. But if you stay nose-down in the code instead of working on getting attention, you're building the second case. Are you sure the market even exists? (p. 71)

This article gives much more detail (p. 9) about what you should be doing, and even how to find those potential customers to interview.

Here's a coder-centric way of thinking about all this more generically: When you tackle a large development project, do you tackle the high-risk, inadequately-understood modules first, or leave those to the end? First, of course, because you understand the rest and you need to solve the unknown problems while you still have time to pivot and re-plan.

This is the same thing, it's just that in a business it's the attention, marketing, positioning, selling, defining part that's high-risk and inadequately-understood, and *all of the coding and design* is low-risk and well-understood in comparison.

So force yourself out of your comfort zone. You'll also do coding and design, and that's fine of course. But force yourself to *mostly* do those *other* things that create a valuable business.

Put down the compiler and talk to customers (p. 1541).

PROOF FROM YOUR PEERS

The above was written in 2013; in 2024 engineering founders still won't listen, as was on display with a tweet from the great Rob Walling.[648] *Here's that tweet, and many successful bootstrapped founders concurring, and lamenting that still no one listens.*

The Twitter thread has far more responses, all just like these.

Can you learn this lesson today, without having to "find out for yourself?"

Just this once?

Thanks to Simón Muñoz[652] *for translating into Spanish.*[653]

I'm so naive

I'm from a software engineering background and I'm very comfortable in that space. I thought building my own SaaS would be super easy. I write code every day, it's just another project: build it, launch it, boom.

Can't believe how stupid I was.

Turns out it isn't enough to implement an idea and put it online after all. Customers aren't going to magically appear out of thin air. Who knew.

SaaS sales and marketing is a whole different thing to implementing code. Like day and night, really.

Rob Walling
@robwalling

Oh developers...they're gonna develop.

I honestly feel for this person.

Still surprises me that 14 years of @startupspod and 13 years of @MicroConf have barely made a dent in stories like these.

Still much work to be done 😅

credit[649]

Daniel Nguyen ⚡ @daniel_nguyenx · Aug 13

I came to the conclusion that this is a founder's rite of passage. The trick is to make this mistake very very quick I think (launch fast, fail then tweak the mindset to market first)

1 15 556

Rob Walling @robwalling · Aug 13

Do you think someone could hear the message from me or a podcast guest and learn from that?

I know some folks have, so this isn't theoretical. But I'm wondering if most people don't hear this message at all, or if they hear it and don't believe it.

6 9 516

Daniel Nguyen ⚡ @daniel_nguyenx · Aug 14

I think yes. But then many smart developers I know are also very stubborn.

I think they do hear about this, but they don't believe it. Or don't want to step out of their comfort zone (building).

2 147

Miguel Sarenas @MiguelSarenas · Aug 14

Yeah I think everyone goes through this phase and figures out the harsh reality haha

2 30

Klaas @forgebitz · Aug 13

building is the fun part, it's also only around 25% of your job

4 224

Eddie Forson @Ed_Forson · Aug 14

Most developers turned founders go through this phase.

I read about it before, told myself I would not make this mistake, and yet still made it 🤦🏿.

Some lessons need to be experienced, not only read about.

3 45

Adam Wright @adamdenverco · Aug 13

It's more of a "we all have to learn this lesson the hard way" problem.

2 89

Aaron Edwards · 🛠️ AI @UglyRobotDev · Aug 13

Been there done that with my first product @infiniteuploads !

31

Stefan Wirth ✓
@NafetsWirth

Honestly?

I read “start small, stay small” and “e-myth” pretty early on and I thought I knew.

But I didn’t.

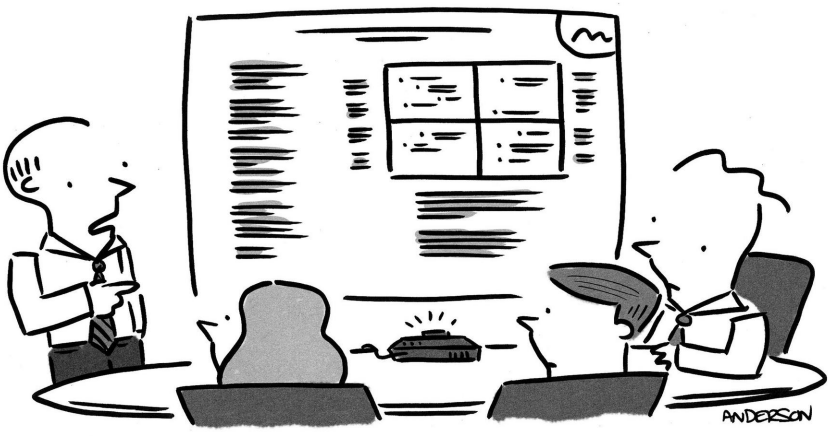
It was always in the back of my mind but avoiding discomfort can go a long way.

Some lessons need to be paid in blood to truly understood.

12:54 PM · Aug 13, 2024 · **279** Views

Chapter 42:
Selecting the right product metrics (KPIs)

VALUE CHAIN · TIME & CONTROL
HIDDEN & EXTERNAL · CHECKLIST



"Thompson, I need you to redefine key metrics through dynamic optimization alignment. The rest of you, figure out what that means."

Executives want financial outcomes, strategists want systemic impact, managers want team accountability, teams want credit for executing work, planners want to track progress, ops wants to know that systems are stable and secure.

How do we select metrics that satisfy everyone?

Much blood has been spilled on this topic. Frameworks range from poster-sized interconnected networks of boxes and arrows,[655] to reductively selecting a single North Star metric[656] to rule them all,[657] to cascading goals like OKRs.[658] An organization should pick whichever framework is most likely to be adopted and honored.

I used to believe that "one true metric" with a smattering of operational indicators was the best way to focus a team on "what matters most." But I've come to believe in a more comprehensive system, that addresses all of the needs outlined in the opening paragraph.

Here is that system. Because all departments can "see themselves" in the result, I believe it makes stakeholders more comfortable, while giving the product team at the center of the maelstrom a practical view, not only of everything they do, but of everything they affect. It allows the team to measure things in the short-run while also maintaining the long view.

METRICS AS A VALUE-CHAIN

A product sits in the middle of a chain of events, executed by the team, customers, and peers across all departments. The first step in understanding metrics, is to plot these events in time, by actor, and by the type of so-called "value" we might measure (Figure 1).

Use coloring to specify which metrics you are maximizing vs satisficing (p. 887). Often "maximizers" are critical for the success of the product, whereas "satisficers" are important operational indicators

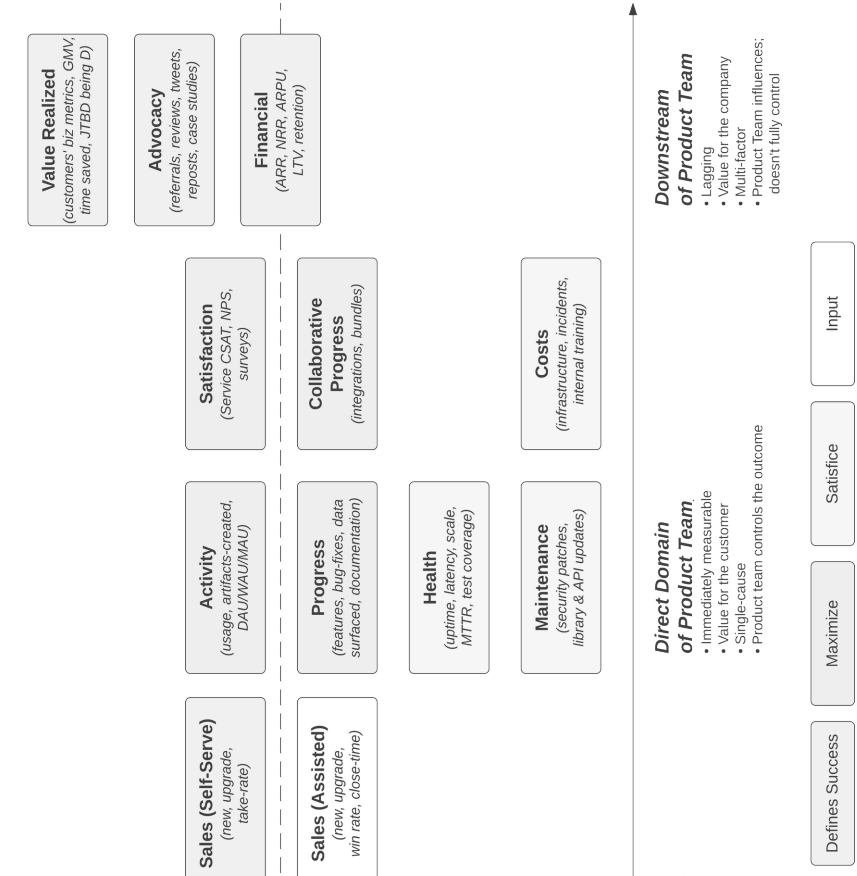

Product KPI Framework
Customer Activity
Company Activity
Higher-level subsequent value
Direct value
Directly visible by Customer
Far from Customer
Marketing
(visits, views, leads, CPC, CAC)
Sales (Self-Serve)
(new, upgrade, take-rate)
Sales (Assisted)
(new, upgrade, win rate, close-time)
Activity
(usage, artifacts-created, DAU/WAU/MAU)
Progress
(features, bug-fixes, data surfaced, documentation)
Health
(uptime, latency, scale, MTTR, test coverage)
Maintenance
(security patches, library & API updates)
Satisfaction
(Service CSAT, NPS, surveys)
Collaborative Progress
(integrations, bundles)
Costs
(infrastructure, incidents, internal training)
Value Realized
(customers' biz metrics, GMV, time saved, JTBD being D)
Advocacy
(referrals, reviews, tweets, reposts, case studies)
Financial
(ARR, NRR, ARPU, LTV, retention)
Upstream from Product Team
• Leading
• Value potential
• Multi-factor
• Product Team influences; doesn't fully control
Direct Domain of Product Team
• Immediately measurable
• Value for the customer
• Single-cause
• Product team controls the outcome
Downstream of Product Team
• Lagging
• Value for the company
• Multi-factor
• Product Team influences; doesn't fully control
Defines Success
Maximize
Satisfice
Input

Figure 1

that can't be ignored, and require attention if they become a problem,* but under normal circumstances shouldn't distract us from what is most important.

Once you have this map, the metrics almost write themselves. Each box has one or more metrics that explains whether "it is happening" or "we're making progress." You could go a step further, adding arrows** to indicate influences or funnels or conversions, adding metrics to each arrow.

THE AXIS OF "TIME, IMMEDIACY, AND CONTROL"

The horizontal axis emphasizes that some events happen prior to activity controlled by the product team, and some happen after. Most metrics frameworks have this concept of "leading" and "lagging" indicators, though some confusingly mix them all together.***

Between those temporal bookends, we highlight that some of a product team's activity can (1) be measured immediately and (2) the

---

* At WP Engine[659] we're very happy with Google's system for managing SLOs,[660] in which "satisficing" KPIs are tracked continuously, but the team acts only when a KPI slips into "violation" territory, as opposed to fielding requests to invest in improving those KPIs. This threshold is pre-agreed with stakeholders, in periodic meetings outside of high-emotion catastrophes, when everyone can soberly decide under what circumstances we will interrupt high-value work to address a critical problem.

** Also called a systems diagram,[661] it could make the chart too busy, especially when "many things affect many things," resulting in a dizzying bird's nest of arrows. Perhaps the simplicity of "boxes only" outweighs the benefit of specifying all the value-flows. Or maybe include only the few, most-important arrows.

*** For example, often top-level OKRs are multi-input, lagging indicators, whereas team KPIs might be immediate, and it's unclear where leading indicators go. This leads to unhappy conversations when teams meet their immediate KPIs, but the company's overall KPIs appear unaffected.

team is in full control (p. 413) of those effects, and therefore the team should be held directly accountable to those things.

This resolves the typical conflict that arises when executives ask "why isn't the team more focused on increasing revenue in the next 60 days" while the team insists "other people don't understand that we're doing a lot of important work." Work can—and should—be measured sprint-by-sprint, whereas revenue is a multi-input, lagging indicator of success. The product team *is* responsible for generating revenue, but it is not the *only* team or actor contributing to that final result, and a change in the product can take a while to show up in revenue; individual features often cannot be directly linked to revenue (p. 1253) at all.

This doesn't make "revenue" less important—indeed, it might be *the* most important metric! Rather, we have placed the metric in context, and understood that it can lag by months or even years,* and therefore isn't a good measure of what's happening right now. If we're successful at our "work" but not our "financial impact," our conversation is naturally directed towards diagnosing that disconnect. If we're not even successfully completing our "work," or if the features that we made aren't being used often, those are immediate facts and within the team's control, and metrics should reveal it, regardless of downstream consequences on revenue. You can't argue against solving for the customer!

Notice that often the "definition of success" also resides in those lagging indicators. This is another traditional cause of confusion, often articulated as "we should celebrate outcomes, not work." It is true that if our work doesn't result in the desired outcomes, we're not finished yet. But, if we believe that the world is inherently unpredictable (p. 193), that not all work will yield a large outcome (p. 221), that outcomes require a combination of execution and luck (p. 449), then we should agree that the job of an agile team is to continuously tackle that

* Blackberry's revenue continued to grow for two years after the iPhone launched; other KPIs were changing far more rapidly, and therefore were even more important to track.

complex challenge, as opposed to expecting every idea to consistently produce the desired outcome.

Seeing how *all* the metrics are performing gives the team and stakeholders the information needed to completely understand what's going on. Which is the whole point of KPIs.

The agile team must honestly and clearly measure *both* direct results *and* lagging outcomes. They are accountable for all of it, and measuring is required for accountability. But "not yet achieving the outcome" is not a complete failure (p. 1261), but rather a learning that will shape the work that will be executed in the next two weeks.

> *"If the result confirms the hypothesis, then you've made a measurement. If the result is contrary to the hypothesis, then you've made a discovery."*
>
> *–Enrico Fermi*

THE AXIS OF "HIDDEN, INTERNAL, EXTERNAL, AND STRATEGIC"

We control what we do in our sprint; we don't control what customers do. The company controls what other teams are doing; the product team doesn't directly control that (much to the chagrin of product managers (p. 817)).

It's useful to draw a bright line between what is external and what is internal. Often completely different people work in these two domains, e.g. product, design, support, and sales working directly with

customers, whereas engineering, infrastructure, and security work inside the company. The diagram helps us appreciate everyone's role, and use the right metrics for the right things.

Furthermore, some internal activities are close to the customer (e.g. releasing new features), while others are far away (e.g. applying a security patch). If all our work is invisible, we have a problem: Customers perceive a stagnant product, competitors appear to be moving faster, sales doesn't have new things to say. On the other hand, if we value only the visible things, we end up with a bad product, with tech debt and unhappy engineers with slow delivery due to an under-invested foundation. The diagram makes this clear, honoring all of these important types of work.

The most valuable, strategic outcomes are often even more distant from the product team, whether because they are down-stream, or because they are second-order effects for the customer. We control "satisfaction" more than self-motivated external "advocacy," (p. 275) yet the latter is clearly not only the ultimate measure of the success of the product, but also drives efficient growth. Product teams should take ownership of creating those outcomes, while not allowing those lagging, multi-factor metrics to be the *only* way we measure progress.

The most valuable thing is for the customer to achieve their own ultimate goal, as defined by the Needs Stack (p. 259). If the customer's business doesn't thrive, they'll stop paying for your software, no matter how good the software is. While of course the customer's business is again a multi-factor, lagging metric, where nearly all the factors are outside of your control, it's still ultimately the greatest form of value. Even if you can't control it, you can notice the attributes of customers who tend to thrive (p. 317), and direct your marketing, sales, and features towards that subset of the market, yielding higher growth and retention, and likely higher profitability.

CHECKLIST FOR GREAT METRICS

From SMART goals[662] to FAST goals[663] to North Star Metrics,[664] there's plenty of prior art on how to pick good metrics. This is my own list.

Defined in normal language
"Customers are using feature X" makes sense. "Total unique IPs which caused at least one event from P, Q, or R to fire in our analytics system over a rolling 14 day period, divided by total unique IPs from the same system in the same period" is a precise way of measuring "using feature X," but it's too hard for normal people to scan and understand.

Defined precisely
In the previous example, you need that technical definition also. Frequently that definition is where we realize either (a) we can't get exactly the metric we wanted, or (b) we have to do engineering work before we get the metric we want. This is especially important with concepts like "cancellation rate" or "cost to acquire a customer" which can be defined in myriad ways; it's often useful to use different precise definitions for different metrics within the same company.

Matches the intent
Often the technical metric doesn't actually measure what we stated in plain language. For example, we intend to measure "User Portal usage" but instead we measure "User Portal logins," which only triggers when someone's session expires. It is common to want to measure P, but P is too difficult or maybe even impossible, so we pick a proxy metric Q. That *can* be OK, but make sure the proxy really does measure the intended concept.

Causes action
If the metric does something differently from what we expect, would we act? Would we re-plan the next sprint, or even interrupt the current one? If the answer is "no," it doesn't belong on our main metrics board. It might belong on an operational board, if it's explanatory, helping us to understand how things are functioning.

Obvious what "good" look like
Not all metrics should be attached to explicit goals, because some are for monitoring and understanding the situation, as opposed to something we're actively trying to change or maintain. However, it should be obvious "what good looks like." That could be as simple as "not changing," or directional rather than specific (e.g. "usage increasing" as opposed to "usage going up 10% month-over-month").

Measures "what is happening" (not "work")
You already track work; metrics shouldn't duplicate or summarize that. Metrics are about "what is happening" around us—the dials on the airplane dashboard (p. 1267), not the actions the pilots are taking. (Exception: Metrics that explicitly measure whether we are completing a volume or quality of work.) If the metric can be moved by, or applies to only one possible course of action, it's measuring work and even presupposing solutions, instead of measuring "what is."

Measured easily
Many metrics are useful, but remain un-updated in spreadsheets because it's too hard to get them. Best is automated; second-best is manual-but-trivial.

Measured frequently
Daily is best. One of the advantages of "rolling N days" is that you can update it daily, yet still think in units like "week" or "month" if that's sensible. Caveat: if the number doesn't naturally change frequently, then it's not important to measure it frequently. Still, in that case you won't check it often, which diminishes its value.

Stable definition and applicability
Measure things whose definition is stable over time, ensuring that any observed changes are the result of deliberate actions or environmental shifts, rather than random fluctuations or alterations in the nature of measurement itself. This stability allows for meaningful month-to-month comparisons and more accurate assessments of strategies and outcomes.

Uses common definitions when possible
It's tempting to invent your own metric, even when there are so-called "industry standards" or "best-practices." Standard definitions might not

be appropriate for your business, or violate one of the other rules above. Inventing your own language is also a form of team-cohesion. However, reinventing the wheel makes it harder for others to understand what you're doing and precludes using benchmarks to see whether your metric is "good" objectively. There are even metrics where special definitions hide the signal; sometimes these are even used for deception.* Don't do that.

Signal at least 2x stronger than noise
Metrics often vary for reasons unrelated to the underlying signal.

Monthly revenue is like this—new, upgrade, and cancellation alike. Calendar months vary in days by ±5%. Furthermore, daily numbers can vary by 2x between a week-day and week-end, and calendar months vary in the number of weekdays by ±10%. So, if you're tracking something like new revenue per month, even a real change as large as 10% is the same size as the noise, so you can't actually tell if there was a real change.

Sometimes a clever alteration to the definition can remove noise. For example, "new revenue over rolling 28 days" eliminates the two factors just mentioned, and is fairly close to a calendar-month worth of revenue.** Or if noise is infrequent, something like "median" or "95%tile" can ignore outliers.

But often noise is less predictable. In that case, it's useful to ask how much is noise, and whether it's so much that the metric isn't useful. If you don't know how much is noise, it's probably a bad metric.

The team actually cares about it
Whenever management requires a KPI that the team doesn't believe in, doesn't respect, doesn't care about, the team always wins. You're lucky if six months later the number is even being tracked; certainly they won't be taking action on it. The team has to *want* to depend on it, look at it weekly at least, and change behavior if it hits some threshold. If you just

---

* Common examples are revenue-recognition, classifying costs in or out of COGS/GPM, and conveniently leaving certain costs out of EBITDA.

** To make it an "average calendar month," you can multiply your trailing 28-day figure by `52/48`, which scales it to the average number of weeks in a month.

assign it, or if there's a sense that the number is meaningless or unfair, they simply will not react to it at best, and decrease morale at worst.

I hope this system will be as useful to you as it has been for us.

Many thanks to Jaakko Piipponen,[665] *Jonathan Drake,*[666] *and Vinod Valloppillil*[667] *for contributing insights to early drafts.*

Chapter 43:
Failure to face the truth

"Turns out my imposter syndrome was pretty much right on."

A primary blocker of progress, from our personal lives to our corporate strategies, is a

Failure to face the truth.

"The truth hurts." Yeah, so we avoid it. The truth is hard to find if we're not looking for it, and we're not. Denial (when intentional) and rationalization (when unintentional) is our normal operating mode. Because we don't realize this, it's an immense, invisible barrier. Whether because we don't like to admit we're wrong (even to ourselves, in secret), or because it will be annoying or painful or career-altering for the truth to be said aloud, we avoid the truth.

Once you start seeing the pattern of "failure to face the truth," you see it everywhere. In almost every meeting, someone is thinking something and not saying it, even though one of the best uses of a meeting (p. 1017) is to unveil and discuss insights. In every strategy discussion, there's a monster in the room no one will name, even though the point of strategy is to identify and then construct a battle plan against the monsters. Written plans and strategies are all optimism and confirmatory data, rather than crystalizing the scary challenges so that we can attack them together.

Those who break through this barrier are rewarded, as many famous books and frameworks point out. It's fun to see this in action, because the examples are individually interesting, and because a totality of instances illuminates the pattern.

Radical Candor

"When you say 'um' every third word, it makes you sound stupid." Not the feedback Kim Scott was hoping for after what she thought was a successful presentation to executives at Google. And it came from Sheryl Sandberg of all people. (Before she joined Facebook).

Now Scott knew, that Sandberg knew, that Scott *wasn't* stupid. Therefore, it was obvious to her that this feedback was meant to help, not "obnoxious aggression" as Scott would later name the same straightforwardness when the motives are to belittle and hurt rather than to coach. And so Scott was grateful for being made to "face the truth." In her words:

> Why had nobody told me for 15 years? It was like I'd been walking through my whole career with a hunk of spinach between my teeth and nobody had had the courtesy to tell me it was there.

The truth is a courtesy; conversely, failure to face the truth is at least impolite, and at worst "ruinous" in Scott's terminology, as when a manager withholds necessary feedback for fear of hurt feelings or being seen as unkind.

Scott is now famous for promulgating this philosophy of direct, honest, empathetic facing-of-the-truth in her book, Radical Candor.[669] It starts with a foundation of personal trust—trust that the feedback-giver genuinely cares for the well-being of the feedback-receiver, and that it is given in the spirit of "facing the truth" and not in the spirit of domination or manipulation or other ill-intent that people in positions of power might visit upon those subject to that power.

We're thankful when a coach admonishes us and shows us a better way; a coach that lied to you, even by omission, telling you everything is fine when it isn't, would be a bad coach.

Five Dysfunctions of a Team

Possibly the most famous book on diagnosing problems in team dynamics and building high-functioning teams, *Five Dysfunctions*[670] by Patrick Lencioni starts with the observation that great teams are built in layers, each requiring the next to be effective (Figure 1).

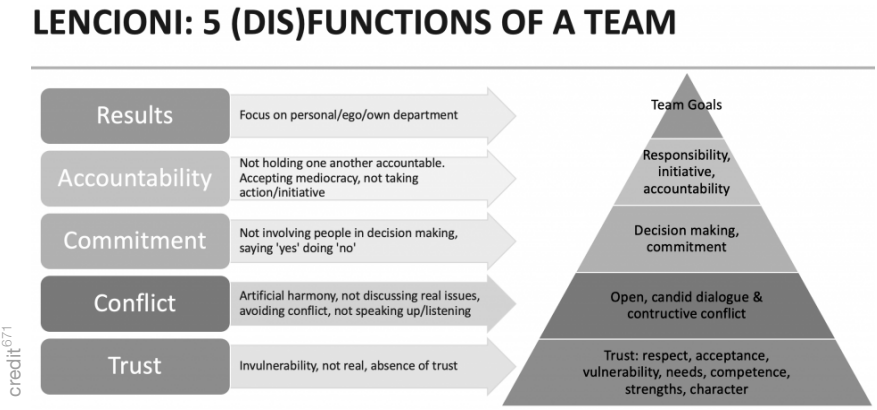


Figure 1

Exactly like Scott's Radical Candor, it is critical that a team be capable of constructive conflict, "discussing the real issues," facing the truth. That requires the team to first build trust, so that it is safe to have conflict.

"Failure to face the truth," together, in the open, prevents teams from being great. We must have enough trust in each other to face the truth, together. If someone can't handle that trust, maybe they're not right for the group after all.

Good Strategy, Bad Strategy

In one of the seminal books defining "strategy," Richard Rumelt lists common hallmarks of bad strategy. Each one is so common, you'll shake your head and laugh (or cry, if you're guilty of these sins):

> • *Fluff*. Fluff is a form of gibberish masquerading as strategic concepts or arguments. It uses [non-specific or] esoteric concepts to create the illusion of high-level thinking.

> • *Failure to face the challenge.* Bad strategy fails to recognize or define the challenge. When you cannot define the challenge, you cannot evaluate a strategy or improve it.
>
> • *Mistaking goals for strategy.* Many bad strategies are just statements of desire rather than plans for overcoming obstacles.
>
> • *Bad strategic objectives.* A strategic objective is set by a leader as a means to an end. Strategic objectives are "bad" when they fail to address critical issues or when they are impracticable.
>
> —Richard Rumelt, Good Strategy, Bad Strategy,[672] p32

His phrase "Failure to face the challenge" is my inspiration for "Failure to face the truth." Strategy must identify and then address the most important and difficult facts-of-the-matter of the market, competitive space, customers, product, and team.

It's scary to say "Our market is shrinking," but if it's true, and you refuse to identify it, if you don't write it down that crisply, if you don't challenge the company to come up with alternatives for how to address it, if you don't build a strategy that expressly attacks it head-on, then it will be fatal. Face the truth.

Lazy language belies a deeper failure

There are traces of "failure to face the truth" even in Rumelt's other bullets. Fluffy language is a personal peeve of mine. In the benign case it is simple laziness—avoiding the work of crafting prose by filling the requisite space on the page with jargon and generic phrases. In the worst case, it belies the lack of having thoughts in the first place.

This problem is rampant in marketing and content-marketing, but sticking with the theme of strategy, it's especially common in "vision" or "mission" statements, theoretically summarizing the company's as-

pirations, but often just a non-specific blob that also applies to any other company in the space, e.g.

> A leading provider of website development for businesses of all sizes.

Well, that was easy! Easy because it says almost nothing, and as a result, it's wrong. Wrong because it doesn't develop websites both for coffee farmers in Ethiopia and also for Tesla's launch of their next vehicle ("all sizes").

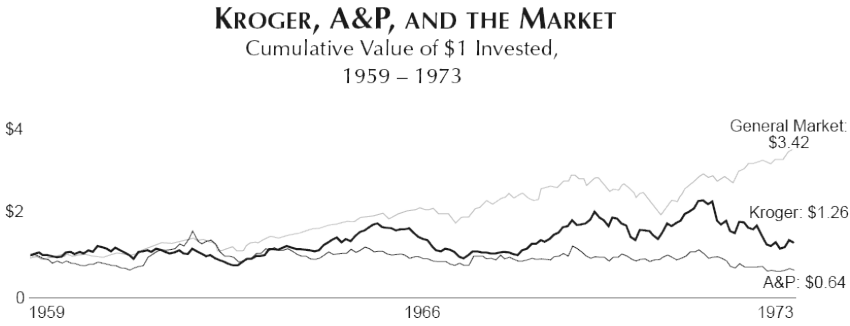
Perhaps it's mere laziness; after all, the only words that communicate anything about what it *is*, is "website development." Or perhaps the company refuses to "face the truth" of what it really does, and who it really serves, where it's really strong but also weak, afraid to say "no" to any potential customer. What if, instead:

> We are boutique artisans chosen by discerning businesses who demand the highest-quality, completely unique web experiences.

Sounds like a small company ("boutique") who nevertheless charges a lot (also "boutique"), but delivers amazing work ("discerning" and "unique"). By saying "no" to people who don't want that, you get to say "yes" to interesting projects at profitable rates, even beating out larger competitors who you can argue are just "factories that churn out the same website with different colors." But only if you face the truth.

Confront the brutal facts

In Good to Great,[673] one of the most-cited books on the formula (if such a thing exists (p. 449)) for successful businesses, Jim Collins frequently returns to the story of A&P and Kroger—two companies alike in dignity in fair 1960s America, where we lay our scene—yet Kroger

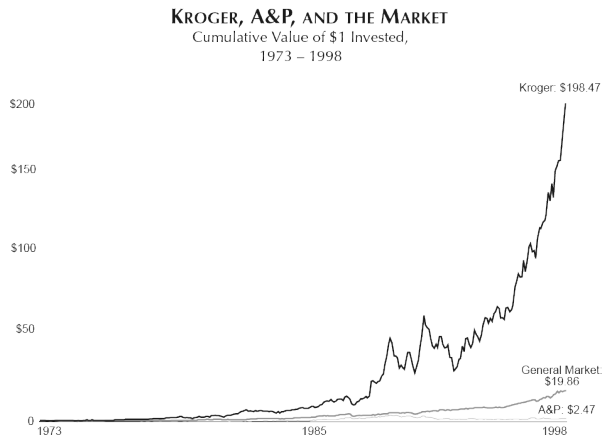


Kroger and A&P grew about the same for twenty years, both under-performing the broader market. Grocery stores are hard.
(Source: Good to Great)

catapulted past A&P because, in Collins's words, only Kroger was willing to "confront the brutal facts."

Collins tells a fantastic story, abridged here:

> A&P stood as the largest retailing organization in the world and one of the largest corporations in the United States, at one point ranking behind only General Motors in sales. Kroger, in contrast, stood as an unspectacular grocery chain, less than half the size of A&P.
> ...
> A&P had a perfect model for the first half of the twentieth century ... cheap, plentiful groceries sold in utilitarian stores. But in the affluent second half of the twentieth century, Americans changed. They wanted ... bigger stores, ... fresh-baked bread, flowers, cold medicines, forty-five choices of cereal, and ten types of milk. ... and they wanted to do their banking and get their annual flu shots. In short, they no longer wanted grocery stories. They wanted super-stores.
> ...
> Here's what's interesting: Both Kroger and A&P were old companies heading into the 1970s [Kroger 82, A&P 111]; both companies had nearly all their assets investing in traditional grocery stores; both had strongholds outside of the major



Kroger exploded with success after "confronting the brutal facts," while A&P continued to falter.
(Source: Good to Great)

> growth areas of the United States; and both companies had knowledge of how the world around them was changing. Yet one of these companies confronted the brutal facts of reality head-on and completely changed its entire system in response; the other stuck its head in the sand.

Collins goes on to detail this last sentence—how both companies were fully aware of these changes, A&P even opening a test store under a different name, which was a "super-store" that didn't even sell A&P items, which outperformed their standard stores. Not wanting to face those facts, they shuttered the test store and went back to business-as-usual. Meanwhile, Kroger made the same experiments, found the same results, and decided to pivot the entire company to become the modern-day supermarket.

A&P saw the truth, but failed to face the truth. Whether it's Jim Collins analyzing dozens of companies or Rumelt with his experience with dozens of strategies, the pattern is the same: Face the truth, or be killed by it.

When it's hard, we avoid it

It seems obvious that avoiding the truth would lead to bad outcomes, so why do we do it? Because it's hard. It's not the only obvious thing in our lives that we avoid, solely because it's hard. Diet? Exercise? Giving feedback? Working on our relationships?

It's hard to tell the truth to someone's face. It's hard to realize that your industry has completely shifted, and it's *really* hard to say that out loud in front of the whole company. It's hard to say "no" to a customer when you have bills to pay, and it's hard to make a major strategic choice, because what if you're wrong?

That's a good explanation for failure to face the truth, but it's not a good reason.

Face the truth.

> *"Go ahead. Faith will follow."*
>
> *–Jean-Baptiste le Rond d'Alembert, encouraging early practitioners of calculus, even though rigorous proofs of its legitimacy were still a hundred years away.*

Chapter 44:
The Startup Drake Equation

THE DRAKE EXPLANATION · DRAKE EQUATION
OVERWHELMING ADVANTAGE · MARKET SELECTION
FOUNDING TEAM · DESIGN OUT WEAKNESSES
PLAY ASYMMETRIC GAMES · HIGH-RISK FIRST

"Things went from bad to worse,
but we're hopeful now that we're
doing badly again."

Most startups fail, even when the founders are smart, driven, passionate, capable, and are solving a problem that people really would pay to have solved. Why?

We already explored the primary causes of startup failure (p. 379) and how to avoid them. If you know where you're going to die, don't go there.

However, all that notwithstanding, failure remains the most common outcome. Perhaps "why do startups fail" is the wrong question—startups fail by default; we don't need fancy explanations. The question is: Why do they ever survive?

And: Why *specifically* are they default-dead? Is there something we can learn from that?

THE DRAKE EXPLANATION

Frank Drake created his eponymous Equation[675] in 1961 to guide discussions at the first meeting of SETI (the Search for Extra-Terrestrial Life). It became a famous a way of estimating how many alien civilizations we should expect to see in the night sky:

$$N = R_* \cdot f_\mathrm{p} \cdot n_\mathrm{e} \cdot f_\mathrm{l} \cdot f_\mathrm{i} \cdot f_\mathrm{c} \cdot L$$

In English: There are billions of stars (R_*), some fraction of which have planets (f_p), some fraction of which are habitable (n_e), on which life sometimes forms (f_l), and sometimes becomes intelligent (f_i), and transmits detectable signals into space (f_c), and have been doing so for long enough for us to see it (L). So that's how many aliens civilizations we should see (N).

Or in modern Marketing language:[677]

credit[676]

Frank Drake, Cornell, 2017

> The galaxy produces many warm leads, but every step of the conversion funnel is brutally leaky, so it's hard to convert to a sale.

Of course the most salient fact about detecting alien civilizations is: We haven't detected any alien civilizations. Even though our telescopes provide that the first few values of the Drake Equation are astronomically large. So that means one or perhaps all of those conversion steps are vanishingly improbable.

Startups feel like this too. Countless side projects are started each day, some fraction of which are intended to become money-making endeavors. While the success rate isn't as low as alien civilizations apparently are, perhaps 999 out of 1000[678] drop off the chain of probabilities—failing to create a venture where the owner quits their day job and brags to outwardly-supportive-but-inwardly-jealous Twitter "followers" about achieving Product/Market Fit (p. 335).

Startups face a chain of risks, or as I like to say, a chain of "ands (p. 1277)"—many things all have to go right. Of course "all things" rarely go right simultaneously; this is why startups typically fail.

credit[679]

"Hey! You there! I said take us to your leader! Hello?"

THE STARTUP DRAKE EQUATION

Here are just some of the factors in the Startup Drake Equation, the failure of any one of which is terminal:

- Product that people (really!) want to pay for
- Able to grab those people's attention amidst the noisy Internet
- Pricing that those people will accept (and that is greater than your costs)
- Competitive and distinctive enough to be chosen
- Able to build the product as promised by the home page
- Sustained value-delivery months and years later, so customers stay and keep paying
- Able to fund the venture, either through early profits or fund-raising
- Able to work well with co-founders (or able do it all alone)
- Develop a repeatable and profitable customer acquisition process
- Able to attract and retain talent

- Able to psychologically handle many years of deep effort, stress, and pain
- Get lucky (capture good luck, dodge bad luck)

It's easy to find examples of failures due to each factor. The non-technical founder who unsuccessfully outsources the product to a consultant (fails "can build it"). The technical founder who builds forever without validating with customers (fails "product that people want to pay for" and "able to get attention"). The tried-but-not-always-true "I had the problem myself, so I built it" (p. 533) origin story, where not enough other people have the problem and the budget and desire to have it solved. Or it was too hard to push through the pain of iteration and pivoting, and anyway the day job pays well, and there's a two-year-old at home, so after six months the founder gives up.

The insights of this model are:

1. The failure of just one element is fatal, so we should spend more time identifying and then addressing the biggest areas of risk.
2. We can analyze and decide strategy in terms of "reducing risk" or "increasing chances" rather than "best ideas" or "unique strategy" or "changing the world".

What follows is how to use these insights to increase your chance of success.

COMPLETELY CRUSH A FEW AREAS TO OVERCOME RISKS AND WEAKNESSES

No organization is low-risk in all areas. But perhaps some areas can be 100%, or effectively "greater than 100%," which then makes up for other deficiencies.

For example, a top-1% engineer might satisfy the question of whether we can build it, but "greater" is wrapping a strategy around the founder of a successful open source project backed by a burgeoning community, which then becomes a unique competitive and marketing advantage, which overcomes deficiencies like not having unique features in the product or not having special skills in advertising. Or, teaming up with a successful influencer who already has distribution, is worth giving up 50% of your equity, because it far more than doubles the likelihood of success. Or, being a renowned expert in some market decreases market risk, both because it's a marketing advantage and because you have insights that others lack.* Whereas if you're entering a market you know nothing about, your education might prove fatal.

I give several examples of this in The Important Thing (p. 1271). You gain both focus and a higher probability of success when you have a singular winning attribute, even when you're terrible at everything else (p. 891). Singular focus and differentiation can overcome deficiencies elsewhere in the Startup Drake Equation:

> Even better than "different," is to be extreme in that difference. Not just a minimal UI, but so minimal it works on the command-line. Not just great design, but so remarkable people buy it only for that, and it's written up on designer's blogs. Not just a new algorithm that solves an old problem, but one that uncovers new things that no one else does, even at the expense

* Although be wary; your experience could be blinding you (p. 1427).

"The rejection rate for this kind of organ transplant is quite high, so we're going to cram three or four in there to better our odds."

credit[680]

of missing things that others catch. You can't do this for all aspects of your product and business—indeed, even a single one is already powerful—but extremity is how you maximize the power of the few things that make you special.

SELECT THE EASIEST PRODUCT/ MARKET/CUSTOMER

We typically think about "target market" with a success-oriented question like: "Who would be delighted by this product?" But a risk-oriented version of this question is sometimes easier to answer:

What would be the *easiest* customer segment for us to target?

Here "easy" doesn't only mean that you *can* do it, or that it's fun, but also that it is profitable. As a negative example, indie hackers* enjoy selling to their peer indie hackers, but indie hackers have no money and usually go out of business in less than a year, so that's a terrible segment. Instead, stable small businesses like dentist offices have large budgets, rarely change software, and last for decades; this is a better market.** Large, growing markets are better still, because "large" means are many niches (p. 317) in which to get started, many adjacencies (p. 793) to expand into later, and incumbents (p. 295) are fighting over new customers, not focussed on new entrants who aren't big enough yet to be worrisome. Big, growing markets are "easy" in this sense.

Koan #44

The bootstrappers

As an army of 10,000
battles another army of 10,000...

...three bandits sneak into camp,
stealing enough to feed themselves for years.

They are ignored; retaliation is low-ROI.

---

* Found mainly on Twitter but increasingly on Bluesky, the quintessential "indie hacker" is a solo founder, who never wants to hire even one employee, who can build software without assistance, who values freedom, flexibility, and autonomy over maximizing money or traditional prestige. They therefore build simple (but hopefully delightful) products, at low prices, designed to be profitable without scaling.

** Better for sustainable revenue, but difficult to sell, as dentists are hard to get on the phone, and aren't usually in the market for new software. Nothing is easy!

The other key factor is in the phrase "for us". Enterprise customers might be willing to pay gobs of money for a decade, but a new company run by a single person will not be able to deliver the complex software, integrations, governance protocols, and professional services that even one Enterprise customer demands, therefore "Enterprise" would be a risky segment choice *for them*.

You must select low-risk markets that are easy for you to address. You could pick a different word than "easy"—lucrative, growing, profitable—but I like "easy" because it keeps things personal. Do you think it will be easy? If so, you're wrong—it will be harder than you think, but still possible. Whereas if you already think it will be difficult, it's also worse than you fear: impossible.

FOUNDING TEAM

When a co-founder is a top-1% engineer, execution risk goes close to zero.* When a co-founder is a top-1% growth marketer, and if the market exists, getting-attention risk goes to zero.

This is one reason why investors like two founders—"one to build it, one to get rid of it" as we used to say at ITWatchDogs. It's not just

* Except in AI (p. 419), where the normal rules are different, and the world's best engineers often in fact aren't making it work.

"getting twice the work done," and not just "someone to commiserate with," but rather it's because you might dramatically improve two different variables in the Startup Drake equation. (Two technical cofounders is far less interesting.)

DESIGN OUT WEAKNESSES

You have weaknesses (p. 569), which creates risk across the Startup Drake equation. But, you can make choices that slalom around most of your weaknesses, making them irrelevant, rather than forcing you to do the impossible: Become great (or even good-enough) at seven different things.

If you're creating a startup on the side, while you hold a day job and a two-year-old, then you should serve an audience who doesn't want tech support, or at least is fine with a 48-hour response-time. That might have implications on how complex the product is, how intuitively it is designed, what customers expect of it, and its price (p. 515). Rather than seeing those as negative constraints, instead realize that this is a superior strategy and product, because it avoids a weakness (amount and consistency of time-available). Indeed, besides solving for the weakness, there are benefits: Your profit margin is higher (because you don't have support costs) and you can everywhere in the world (because neither language nor timezones are a barrier) and it will be delightful to use (so the only communication is asynchronous praise on Reddit). This can even be done at scale (GMail, Facebook, Twitter, most hardware products). Suddenly a negative "constraint" looks like an insightful advantage.

Or if you're a terrible designer like me (p. 853), it would be high-risk to make a product that must appeal to designers or marketing agencies, i.e. people who value and appreciate great design. You'll do

IF ENGINEERS WERE TREATED LIKE DESIGNERS credit[681]

just fine selling to infrastructure engineers or backend Enterprise systems managers.

Or if you're terrible at marketing, you could create a collaborative product where people have to invite other people in order to use it. While that mechanism is difficult to get started, it means even poor marketing can result in a growing, healthy company. (It's funny that a "viral"[682] company is also "healthy.")

Or if you cannot write code, but you are good at selling yourself and solving a class of valuable problems, you could avoid the world of software (whether as-a-service or not), instead creating a "productized service," in which you sell services, but fulfill that service at low internal cost (and therefore high profit) thanks to your "secret sauce" internal workflows, spreadsheets, and no-code software, which only has to be good enough for your own employees to use.

By acknowledging your weaknesses as fervently as you're proud about your strengths, you can increase the chance of success by avoiding them, and embracing the knock-on implications.

I cover this in detail in my article on Pivot Points (p. 569).

PLAY ASYMMETRIC GAMES

As the old joke goes: "The probability of anything is 50%: Either it happens, or it doesn't."

In truth the probability of success is typically unknown—the variables in the Startup Drake Equation don't have clear values (p. 997). Although that means there will be failures, it also suggests there will be some successes. If the positive magnitude of the few successes exceed the negative magnitude of the failures, we are a success overall.

After all, in most startups, most things are mostly a disaster most of the time. New people joining the company often say, "Wow, I can't believe you're doing _____ and yet you're still in business!" They aren't wrong; they're unwittingly proving that some things can be so powerfully positive, that it overwhelms deficiencies.

Many examples of this appear in the "Asymmetric" section of What makes a strategy great (p. 504). It's summarized best by Jeff Bezos, who led a company that never stopped taking asymmetric bets:

> "I've made billions of dollars of **failures** at Amazon.com. Literally.
>
> But a few big successes compensate for dozens and dozens of things that didn't work."
> —Jeff Bezos, The Guardian,[683] 2014

On a smaller scale, there are marketing efforts with a fixed cost, but also a fixed maximum upside, like advertisement where you pay whether or not someone signs up. Or there are marketing efforts with a fixed cost, that could generate new customers for years to come, as

with reputation (social media, community forums) or media (successful SEO).

Or there are markets where there are many different ways to win, many niches to try, many routes to customers, many ways to expand from a successful foothold into a wider space, and markets which are stagnant, perhaps shrinking, where every sale is a hard-scrabble against desperate yet monied incumbents.

Pick games where the downside is 1x but the upside potential is 10x or 100x. Then if the realized benefit is "only" 3x, the game is still a success, and hopefully makes up for other weaknesses.

ATTACK THE HIGH-RISK AREAS FIRST

With complex projects, the Risk-First Heuristic[684] tells us to tackle the high-risk things first.

If you start with low-risk tasks you'll surely succeed, but the high-risk tasks remain unresolved, and remain high-risk. Nine months later, when you finally tackle the high-risk areas, you might discover the entire venture is unworkable. Even if not fatal, new insights means rework, which indicates you've wasted time. It's smarter to address the risky things first, learn from them, change because of them, and complete the rest of the project informed by them.

The Startup Drake Equation lists potential risk areas. Even after designing our product, market, and strategy to circumvent or remove risks, some high-risk areas will remain. Circle those, and address them first.

So if we're incorporating AI (p. 419) into the product, make sure that actually works first, in real life, because usually it doesn't* (at least not as of this writing, as thousands of self-styled "AI Startups"

have discovered). Or if we're unsure what potential customers would actually pay for (as often they *have* a problem but don't want to pay to solve it), we should find that out (p. 239) before writing code. Or if we've never marketed a new product before, we should make advertisements and landing pages that lead to a waiting list, to make sure we can garner attention before building something that no one will ever see (p. 1541). Or if we've never developed a reputation that enables us to talk to customers, we should do things to create those conversations (p. 683).

All startups are risky, and even with this model in hand, most startups will still fail.

But, by clearly articulating the risks, reducing some risks through the founding team and selecting the market and product that is right *for us*, solving one or two so well that they overwhelm other risks, constructing a strategy that designs around things that would otherwise be weaknesses, and attacking the remaining high-risk areas first, you can dramatically improve the chance of success.

Those aliens are out there! Maybe you're one of them.

---

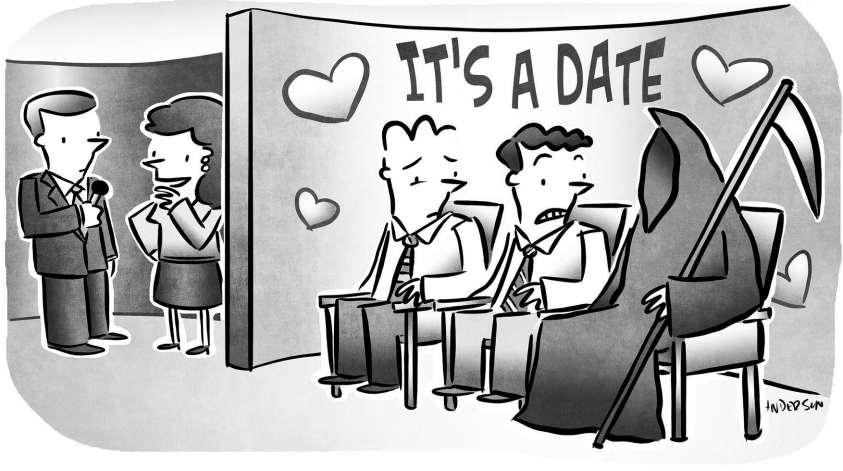
* As an executive once quipped to me twenty years ago, explaining why a new product line had failed: It was a platform portability problem—we couldn't port it from PowerPoint to Java.

"We'll probe you in a bit, but first I'd like to get to know the real you. Your hopes, your goals, your dreams… Who is Fred Wilson? What makes him tick? And *then* the probing. Lots and lots of probing."

credit[685]

Chapter 45:
When you have nothing: How to find potential customers to interview

GROUND RULES · TECHNIQUES
MORE RESOURCES



"Again, that's like *super* dark, Bachelor Number Three, but I have to admit I'm intrigued."

No matter how many features, no matter how great the UX, no matter how cool the AI, if people don't care, or don't want to pay for it, or want it to work differently, then your company will fail.

You know; you also know you should interview potential customers *before* you start building.

But **how do you find these potential customers?** How do you get them to talk to you, when you have no brand and no product?

GROUND RULES

General advice applying across all techniques:

1. **Only interview your ICP** (p. 317)**.** Early on, you're not entirely sure who that is—fair enough—but you have a general sense. Family and friends will tell you "that sounds cool" and cannot tell you whether *actual* potential customers agree, and want to pay. Go where the ICPs are.
2. **"Good enough" is good enough at first.** Best is an hour-long interview as described in the guide referenced above. But beggars can't be choosers, so it's OK if it's partial or otherwise not ideal. If you run a partial interview, and the other person seems interested, **always ask for a follow-up** so you can go deeper; that will be 10x more valuable.
3. **Frame it as asking for advice.** People hate being surveyed but they love getting asked for their opinion. Prostrate yourself ("it would mean so much to me to have your advice") and compliment them ("because you're an expert" / "... told me you're the go-to person for this" / "I've really loved your article on ...").
4. **Make it trivially easy for the other person.** Proactively send a calendar-scheduling-link so they can find a time that works for

them. Have lots of time slots open so it's not a hassle. Minimize back-and-forth communication to set up the meeting. Respect their time; don't create barriers.

5. **Expect a low success rate.** Cold-calling is the worst. It's soul-crushing for you and annoying for others. Reaching out to strangers is a form of cold-calling (though many of the techniques below warm things up a bit first). A 10% response rate would be fantastic. Don't take rejection or apathy personally. You deserve this. (p. 457)

TECHNIQUES FOR TALKING TO POTENTIAL CUSTOMERS

Techniques that I and other entrepreneurs have employed successfully:

Ask for referrals

Ask your entire network to introduce you to ICPs. "Your entire network" contains almost no ICPs, but *they* probably know some. Have a tight description of the ICP so they immediately think of the right people. Use family, friends, social media groups, LinkedIn, professors and students, social and religious groups—anyone and everyone might know an ICP to refer you to.

Then, **at the end of every interview, ask who else you should talk to.** Try to get two or more. People will agree but then not follow up; here's how to recruit them to be part of the journey:

> "I'm trying to have 20 conversations like this, to give this product a real shot. It's my dream to create something people genuinely love. It would mean *the world* to me if you could help me by intro'ing me to a few other people who are [my ICP]".

If you average one per interview, you have an infinite supply of interviews.

LinkedIn

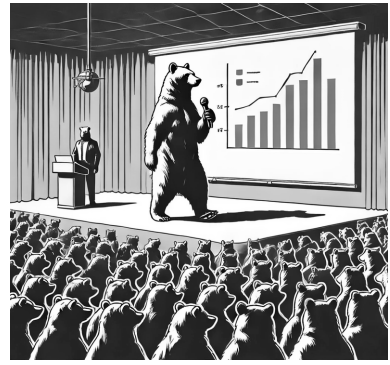
You can search by title, and sometimes get introduced through connections. See, you thought LinkedIn wasn't a useful a social network, but it's ideal for this quest.

The way I got interviews for WP Engine is by using LinkedIn to find people who had the title and industry I was targeting (web developers using WordPress), and asked them for an hour of their time to chat about a new startup concept for whom they are the ideal customer. Out of 50 requests, 40 agreed, and 30 eventually became customers.

How I got such a high hit rate:

1. In my initial outreach, I said: "I will pay any fee you feel is fair for an hour of your time. I'm not asking for a donation; I'm genuinely interested in your expertise and opinion." This showed respect, both for their knowledge and time. I was shocked at the response: Only one person out of 40 asked to be paid! Reciprocity works.
2. I researched each person quickly—who does their business target, where are they located, was there anything in recent blog posts I could personally connect with or comment on—so that the outreach was personal. Yes, this takes time, but the conversion rate speaks for itself.
3. I tried to get an intro from an existing connection. People will often help, even if they don't know the target person too well; write the InMail for them, where you're polite and don't over-sell yourself, and talk like a normal person,[687] not like a robotic marketer (p. 627).

Networking Events & Conferences



People go to network events to talk, and the subject they enjoy talking about the most is themselves. Good! You want to talk about themselves too. Of course the event needs to attract your ICP so you're not learning from the wrong crowd. Some of your questions will be more natural in that sort of environment; have 3-5 of them in mind and pick up some lightweight information.

Bonus: Have enough useful things to say to earn a speaking slot. Speakers have instant credibility during the entire conference; it will be many times easier both to start conversations and earn longer meetings later. Local meetups and larger conferences alike have a constant appetite for fresh speakers.

Bonus: Do something creative and noteworthy at the conference. At Fincon, Noah Kagan[688] gave his speaker fee[689] to the audience ($50 per person), telling them to use it to kick off their entrepreneurial journeys (the subject of the course he was selling).

Forums: Reddit, Discord, Slack, Facebook Groups

Selling your product in a forum is a no-no, but fortunately "customer development" is not selling; it's learning about the lives of others. There will be multiple Subreddits / Slacks / Discord servers / Facebook Groups where your ICPs already congregate.

Use posts to ask your questions. You can probably ask all your questions over time. One post per question, and don't flood the system. Responses will be discombobulated—responses from a single person aren't connected like they are in a real conversation—but this is still a great way to survey and start honing your hypotheses.

Reddit, Slack, and Discord tend to work better in real-time, so lurk to discover when they are more active, both for posting new questions and to establish your reputation by responding to others. Take the time to provide high-quality answers to others' posts and comments; communities are give-and-take, not just take. The better communities will have moderators enforcing this, and rightly so. This is a long term relationship that will carry you through the coming years, not a hit-and-run. Here's a mini-guide[690] from Jordan Malone[691] on how to do this.

You might be able to do something similar with Hacker News,[692] Quora,[693] or even Craigslist,[694] but those are general forums where it's difficult to target ICPs, and almost impossible to create long-term relationships. Conversely, sometimes there are small communities with independent websites, who would welcome an additional member, or even entertain inexpensive, tasteful advertising for you to reach their members.

Patrick Gallagher
@gridpane

Facebook groups FTW. In the early days probably 95% of our new customers came from FB groups. A lot of what we built came as a direct result of seeing the problems that people were sharing in places like The Admin Bar.

Skool is getting more and more popular in this context but FB groups are many orders of magnitude larger and more established.

credit[695]

Figure 1

Don't be too quick to ignore the old-school forums like Facebook Groups, even for modern software (Figure 1).

Ask people who are asking for help, or complaining about the problem / competitor

People rant online all the time, especially in social media (don't forget YouTube comments), forums (e.g. Reddit, Slack, Discord, Facebook Groups), and review sites (e.g. G2, TrustRadius, Capterra). Set up searches for your area, the problem, and competitors. When someone complains, reach out, probably by finding them on LinkedIn and referencing their review. Jordan Malone[696] has a mini-guide[697] for this too.

Many products have a feedback site where customers can vote on pain points or features. Careful though: Your product will be different, your customer segment could be different, your goals and trade-offs will be different, and your product isn't already mature. Still, trends can be informative, and individual people who are especially verbose are more likely to want to talk to you about it also.

Even better: Look for people who have taken action, i.e. evaluating other products, switched products, built their own workarounds, etc.. It's free to complain, but expensive to act, so they must really mean it.

Or the positive version: People simply asking for help. "How do I _____?" Use keyword-monitoring tools, and then actually help them. Having delivered something of value, you can now ask whether they'd hop on a call to talk about it more.

Buy traffic to a landing page

Set up a one-page landing page asking for people who are interested, then buy advertising* to send traffic there. Bid on keywords with low competition, even if volume is small; you're looking for a few targeted visitors, not scaling a marketing channel. You will need to A/B test (p. 913) your ads and your landing page. Tim Ferris famously used AdWords to test book titles and market-positioning for *The 4-Hour Work Week*, before he was well-known and had a following; Katt[698] has built multiple products using this method.

Getting clicks gives you a positive signal on positioning and language—useful, but it isn't a customer interview. Your landing page will need to collect emails from people who are willing to talk. You could offer something in exchange—a gift card to Amazon, a donation to some charity, etc.. You might offer "first year is free once the product comes out."

Bonus: Add web-chat, and pop up when someone comes to the site. When they see you're not a bot, you've suddenly become a real human

* AdWords is the most obvious place to start, as it's the most sophisticated with targeting, A/B testing, and budgeting. However, don't forget about Twitter, LinkedIn, or even Facebook; go where your audience already is (p. 1027).

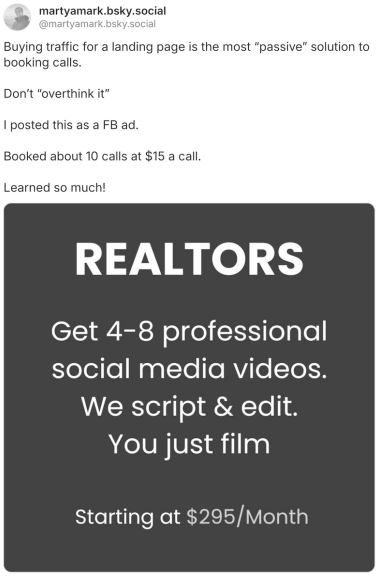
being instead of a cold automated form. Explain why you really value their advice, offer a fee if they're considering but unconvinced, and appeal to the idea of helping a struggling founder with a new venture.

Bonus: Create some content that the ICP really cares about, hidden behind an email address. Then you're giving them something valuable, sometimes earning the right to hop on a call.

Bonus: Put your self-scheduler link right on the page, or like John-Paul Anderson did for his startup,[699] minimize conversion steps by making the scheduler *be* the landing page, including the fee you are paying the participant for their time.

Ask someone who knowledgeably talks about it

Anyone who is already blogging / newslettering / social-posting about this problem has demonstrated that they like talking about it; of course best is if they are the ICP. Ideally engage them first by thoughtfully



credit[700]

responding to things so they have some connection to you. Then a chat might be fun for both of you, especially if you offer to share the results of your total findings with them. It doesn't matter how many followers they have! Indeed, people with fewer followers are more likely to engage.

They might also be willing to ask their audience on your behalf. Maybe later they would promote your software, especially with an affiliate link. The smaller the audience, the more likely they are to work with you; so long as the niche is targeted, this is not a bad use of your time when you're just starting out.

(Ironically this is partly how Reddit got started... and now they're a primary source in their own right!)

Offer to write guest-posts for newsletters and blogs. These have to be fantastic though; significantly better than their median post. It works better with sites that typically feature multiple authors; worse with single-author blogs who never feature guest-posts; you're not going to be the first exception.

Work with an influencer

Influencers have an audience, and are always looking for excuses to do something interesting with them. It doesn't make sense for them to spam their audience for you, so get creative with how you could help the influencer:

Maybe you will gather your results, share the summary with them, and they can make an interesting post. Maybe they'd be interested in promoting the product when it comes out, with a revenue-split. Maybe you can use your prototype or expertise to give them a report that is genuinely useful to them, or you can offer up a truly great guest article / newsletter / social-post.

Pay for meetings

Websites like User Interviews[701] (and maybe Fivvr[702] or UpWork[703]) are full of people who are willing to talk for money. Beware: You need a strong ICP filter. It could be that your ICP isn't there at all.

Expert Networks like GLG[704] and Third Bridge[705] connect you with verified industry experts and operators. While it costs more money than most of the options here, you can often interview Directors, VPs, or even C-level officers who you may not be able to access any other way. You can have candid conversations about budget, decision-making processes, and competitive dynamics that would be awkward or impossible in other contexts.

Angels & VCs

Whether you want to raise money or not, angel investors and VCs talk to founders for a living. They often have a broader perspective on an industry, and anyway they enjoy this sort of thing. They might be able to give you insights despite not being the target customer, but take everything they say as "ideation" and not "validation" nor "insights," because ultimately they don't know any more than you do. If they did, they wouldn't have such a large failure rate in their investments.

However, they probably know many people you *should* talk to. Having a network is also part of their job. Therefore, even if the initial meeting is mostly ideas and little validation, if they like you they are very likely to introduce you to actual ICPs.

Ask people who are professionally engaged

Your ICP probably hires people. Their job postings reveal their exact pain points - look for patterns in the "requirements" and "responsibilities" sections. Reach out to the hiring managers; they're thinking deeply about these challenges right now and are often willing to chat about potential solutions.

Look for people who backed similar or complementary products in your industry; people who prepaid on Kickstarter or voted on ProductHunt might be more willing to talk to you.

Newly-funded startups are "on to something," have customers of their own, and yet might be excited to talk to another founder; check Crunchbase and AngelList.

Build in public (with others)

There is a strong online `#buildinpublic` community. Twitter used to be the main place; now Bluesky has a strong game and is getting stronger. There's some on Threads too. (LinkedIn isn't very good for this). Even if you have very little following, you can join the conversations that are already happening, often under that hashtag. By commenting (usefully!) on others' posts, and making some of your own, you build enough rapport that you can start asking them for referrals (see above). Some might directly be your ICP, but beware: other people who are just starting out are almost certainly *not* your ICP.

Create a free micro-tool

A great technique both for initial conversations as well as a permanent inbound marketing funnel, create a free tool that is SLC (p. 101)—something useful, simple, well-designed, and complete.

Indeed, one of the first companies to pioneer this method was Hubspot, who also pioneered the modern conception of Inbound

Marketing on their blog[706] and their book, appropriately-named Inbound Marketing.[707] Even at $2B in revenue, the technique is still working, with more than 30 tools now available.

Go on a podcast

There are 4 million podcasts; most are defunct, and most have no listeners. But you don't want listeners, you want to interview someone. Propose to the podcaster that you want to do a special episode where *you* interview *them* on this topic. It should be interesting for others to hear this level of detail; you might need to tune your questions to make it interesting in this context. Of course the podcaster must also be an ICP.

Or, create your own. It's more work, but it can become a long-term marketing channel, as it is for Steve McLeod:[708]

> I have a podcast for the purpose of contacting my ICP and inviting them to talk about themselves for 30 minutes.
>
> The podcast description states that it is for my ICP. Once I had done a few episodes, people with my ICP started hearing about it and would then contact me and asking to be interviewed.
> What about just one line?
>
> I now have a nice waiting list of people wanting to be interviewed.

Cold outreach (email, DMs, social)

The conversion rate is low, but the outreach might be inexpensive. Like the landing page, you will need to A/B test a tight, compelling message, and the call-to-action would probably benefit from an offered reward.

Go where people are (waiting)

It's hard to interrupt someone while they are working, or while they are going somewhere. But people standing in a line have nothing to do. This could be at a checkout line, waiting to get into an event, sitting in an airport terminal, standing in line for coffee, or between sessions at a conference. Even virtual "waiting rooms" like the comment sections of live-streamed events can work; people are just waiting for something to happen.

Sometimes you can interrupt people if you come bearing gifts. Nick Basile used free donuts[709] to talk to university students. Karsten Duus Wetteland[710] found people in the building where he worked; the fact that they would probably bump into each other in future made them more friendly and ameniable. If you already work in the industry that your product idea is in, you can interview people at your company.

PR

For most companies, "PR"—getting articles and interviews from traditional media—is reserved for late in the maturation of the marketing department. At first you're just trying to get those first few customers, then getting one marketing channel to work[711] well enough to establish the company. However, if you are an expert in a topic, and have insights that others would find interesting, and are good at being interviewed, then early PR is a way not only to start establishing your brand, but to gain the credibility and inbound interest to have those

initial customer conversations. You have "given first," and now you can "take."

In the extreme, this can take the form of "stunts," as when Brian Chesky (AirBnB) distributed branded cereal boxes called "Obama O's" and "Cap'n McCain's" during the 2008 Democratic National Convention in Denver to generate buzz and media coverage, or when Alexis Ohanian (Reddit) sent personalized luggage tags with a Reddit alien logo and handwritten notes to the first few hundred people who mentioned Reddit on Twitter.

Make your own events

This is a direct quote from Giff Constable's[712] *excellent book Talking to Humans.*[713] *It's only 80 pages; I highly recommend it.*

> One aspiring entrepreneur wanted to target mothers of young children. She had heard stories about talking to people in a coffee shop, but felt like it was too unfocused. So she tried hanging around school pickup zones, but the moms were too busy and refused to speak to her. Next, she tried the playground, where she figured moms would be bored watching their kids play. This worked reasonably well, but she was only able to get a few minutes of anyone's time. So instead, she started organizing evening events for moms at a local spa where she bought them pedicures and wine. The time of day worked because the moms could leave the kids at home with their partner. The attendees had a great time and were happy to talk while they were getting their nails done.

In another example, Jitpal[714] created an industry event with a panel of experts and C-suites; the audience was also serious people. It was a lot of cost and work, but establishing themselves this way allowed them to have many conversations with ICPs, and also paved the road to initial sales.

You can also create a local meet-up on the topic that your ICPs care about. This works better in a larger city, takes a lot of time, and can last for years. But, it also can become a permanent fixture of your brand for customers and recruiting, and a lot of fun if you're an extrovert. You also don't need to have an existing "following" or reputation to do it; Meetup.com[715] helps spread the word, and multiple people have had success using that site alone.

If all this sounds like a lot of work, that's because it is.

Good news: **You only need one or two methods to work**, especially if you're good at generating new referrals from each interview. And this isn't a process you have to do forever, just enough to either invalidate your idea (p. 845), or to change your idea such that it is validated.

What definitely *won't* help is creating software, adding features, and yapping about nothing on social media. So, put that stuff away, and get in front of some potential customers.

MORE RESOURCES

- How to create and execute customer interviews (p. 239).
- Destroying your excuses for not interviewing customers (p. 1469).
- *The Mom Test*[716] by Rob Fitzpatrick (2013)—probably the single best book on this topic, and compatible with all the articles on this site.
- *Deploy Empathy*[717] by Michele Hanson (2021)—how to find and talk with customers with a mindset that unlocks what you actually need from them.

- *Traction*[718] by Gabriel Weinberg (founder of DuckDuckGo and others) and Justin Mares (2015)—an under-appreciated book on how to get initial sales (or conversations). (Sometimes confused by the other book by the same title, which is also excellent but a different topic.)

Many thanks to Carter Bryden,[719] *Counterplot Consulting,*[720] *Giff Constable,*[721] *Karsten Duus Wetteland,*[722] *Katt,*[723] *Jason Evanish,*[724] *Jitpal,*[725] *John-Paul Anderson,*[726] *Jordan Malone,*[727] *Justin Jackson,*[728] *Marty Markenson,*[729] *Marie Strauchman,*[730] *Nick Basile,*[731] *Omar,*[732] *Patrick Gallagher,*[733] *Steve McLeod,*[734] *Tony Meijer,*[735] *Tino Go,*[736] *and Wim Cools*[737] *for contributing their experiences.*

Chapter 46:
The unfortunate math behind consulting companies

DOUBLE IS NOTHING · WHAT TO DO?
IS IT THAT BAD?

credit 738

"I dunno, lately I'm just not incentivized..."

Consulting can be a great way to fund a startup or make a bunch of cash. It's easy to start; Just pick an hourly rate and jump in. There are tons of online tools for time-tracking, project-management, and billing.

But someday soon you'll notice there's only so many billable hours in the day, and you'll be tempted to expand. Maybe hire an employee for $30 per hour and re-bill them at $60. Easy money, right?

Unfortunately the math doesn't work that way.

Here's what really happens, and a few ways to combat it.

DOUBLE IS NOTHING

Suppose you hire an employee at $60,000/year. There are 2,000 working hours in a year—40 hours per week times 50 weeks—leaving two weeks vacation. So your cost per *billable* hour is:

Nominal Cost: $30/hour

In terms of working days (WD), there's 250 in a 50-week year. In America we have ten federal holidays[739] bringing the annual WD count to 240. Recomputing the cost of the remaining *billable* hours, we have:

20 WD/month, $31/hour

We also have employment tax in America. Rules are complicated and vary by state, but as a rule of thumb you pay 15% in taxes including Medicare and Social Security. That changes the annual cost of your employee from $60,000 to $69,000 with an associated change in hourly cost:

20 WD/month, $36/hour

We've assumed an 8-hour work day, but any owner of a consulting company will tell you this rarely happens. Oh sure, *founders* work 60-80-hour weeks, but not employees. Besides, even if you're in the office 8 hours a day it's hard to be heads-down-on-task the entire time, and consulting is about *billable* hours. It's hard to be billable for 8 hours a day, every day. From what I've seen the weekly billables are more like 36 hours instead of 40. That means you're short 16 hours per month, which is equivalent to missing two full (8-hour) days. With this loss, now we're at:

18 WD/month, $40/hour

Then personal life intrudes. You come in at noon because you had a morning dentist appointment. You take off early to wait for the A/C repair guy. You're sick one day. All very reasonable and it doesn't sound like much, but two half-days and a sick day means another two days lost:

16 WD/month, $45/hour

We've assumed you're able to keep your employee completely busy throughout the year without down-time between projects, but that's unrealistic. If you get two-week gigs, maybe there's a day between them where the timing didn't quite work out or you did some post-contract clean-up. If you land a six-month gig, that's better, except it's hard to time large deals like that to start and end *exactly* when you want, months in advance; in fact it's common to have 2-3 weeks of down-time between such things. Either way you lose at least 10% of your time, and that's assuming a healthy incoming project pipeline, so knock off another two days (10% of the 20-day work-month):

14 WD/month, $51/hour

And let's suppose you want to allocate just a little time for career development. After all, if you expect to hire and retain great people, they need time to keep learning, stay in front of the latest technology, and just have fun. Maybe that means going to a conference or two, or working on an open-source project, or learning some new tech that's needed for work. You see what's coming of course: That's another 10%:

12 WD/month, $60/hour

The true cost of an employee is *double* the nominal cost. If you bill out a so-called "$30/hour" employee at $60/hour, you'll only break even. You really need to bill out at $100/hour to make any kind of profit.

Which is hard, because the client you're billing knows this person doesn't cost even close to $100/hour. And when that client thinks about what's "fair," they won't go through the computation I just did; they'll base it on the person's nominal rate plus a little profit for you. This caps the amount you can actually re-bill before client feels ripped off.

I hear you say: "Yeah, but at $100/hour that's $40/hour profit, or $80,000/year to me! So that's still really good." But consider this: If your underling can be billed at $100/hour, can't you bill at $150/hour at least? And if you worked just 20 hours/week (on average) at that rate you'd clear $150,000 annually without the hassle of hiring and managing an employee. Isn't *that* better?

WHAT TO DO?

So how do you mitigate these issues? Lots of ways, few of which are compelling, but all of which are quite do-able:

Scale

If you had *five* of these employees you'd be clearing $300,000 per year, which sounds better. **Except not** because scaling creates more time and expense:

- The client pipeline is much harder to maintain. Keeping 3-6 projects going year-round is a full-time job in itself—a job resting on *your* shoulders.
- Employee turnover becomes more common, so you're permanently in hiring mode. That in itself is time-consuming and expensive, and it's hard to coordinate new hires exactly when other employees are leaving.
- With six people you'll need office space with the attendant expenses, *or* a significant, possibly international travel budget for quarterly get-togethers.
- With all these new tasks, there's no way you can also manage the projects and client relationships and internal product development, so you'll need a project manager or a sales person or an office manager or *some* kind of help, and all of those come out of your profits.
- **None of these new tasks are fun or creative. It's drudgery, and it's all on you**. Congrats, you're a business owner.

With these new expenses it's not unusual to see that so-called $300k in profits came back to $150k. And now you're doing things you hate doing. Scaling is hard (p. 773).

Charge more

Any amount more you can charge **goes straight to profits**, so even charging a wee bit more (or a *lot* more (p. 53)) makes a big difference at the end of the year.

Often, though, charging more pushes away your existing client base, and you'll have to find a new breed of folks willing to spend the big bucks. These are generally bigger, more dysfunctional companies with 47-page Master Service Agreements and complex, ever-changing requirements (though without an acceptance of "agile" or "lean"). It's a living, it's just a different living than being a Cool Cat HTML/CSS whiz doing fun projects for other entrepreneurs.

This sword cuts both ways—even little discounts crater profitability (p. 895). For example it's common practice to give a 10% discount if the client pre-pays, but although it's great to have cash up-front, that discount comes right off your bottom line. (A better idea is to raise your prices by P%, then offer a P% discount for pre-pay. Then your cash-rich customers are paying the same, but others are creating more profit in exchange for delayed payment—a reasonable trade for both parties.)

Bill more hours

Billing even 5 hours more per week significantly increases profitability.

The problem is convincing your employees to work more, because to them that's "overtime" on a fixed salary. What's in it for them?

One answer: Split overtime billings with them. It's pure profit to you and your employee can earn a significant bonus.

Build a product

Many consulting companies have a few ideas or even working internal projects which they hope to someday turn into money-making products. The usual story is:

> We built this product because we needed it ourselves. It gives us an advantage in consulting because it accelerates our development and enhances our sales pitch.
>
> But surely we're not the only software developers on Earth who would find this tool useful, so whenever we have a lull in client work we work on our product. Someday that will cashflow by itself, at much better margins.

Indeed there are conspicuous companies who grew successful product lines in exactly this manner: 37signals,[740] FogCreek,[741] and Pivotal Labs.[742] Unfortunately, for each of those there are *hundreds if not thousands* who toil away at pet projects which never see the light of day. Some reasons:

- Converting an internal project to a user-friendly one requires drudgery: Onboarding flows, documentation, intuitive user interface, installers, password-reset, and fixing those 200 bugs that internally you've just learned to work around. No one wants to do it, and it's no one's *job* to do it, so it doesn't happen.
- It's no one's job to market or sell the product. No one has the time, and it's quite possible that in a roomful of consultants no one has the skill-set either. So you "put it out there" hoping that the magical world of "social media word of mouth marketing" will bestow eager users upon your sign-up form. But, of course, that's not what happens.
- You have five kinda-cool projects, any of which might be awesome if given enough TLC, but because your attention is spread among all of them (and even then only in fits and spurts between client

work) none gets to the stage of being usable by anyone else. **Lack of focus kills.**
- Billable hours trump product development; it's always second-class, so you get a second-class side-project.
- Thinking that just because the product is "great for us," that there's a market out there. Usually there's not. (Here's how to figure it out (p. 71)).

Most of the problems above can be fixed by **treating the product as a client**:

- The product is listed as a client just like all the rest.
- The product is given a budget, even to the point of a separate bank account (and corporation?) so it's clear how much you've spent on it, and the project pays consulting fees (perhaps at cost instead of full-rate).
- Hours are scheduled and prioritized like other clients, and doesn't slip just because it's not a "real client."
- If the (potential?) income from the project cannot justify the cost, you might consider it a loss and stop completely rather than continue to limp along as a money-losing project.

But ultimately, it's just hard.[743]

Use subcontractors instead of employees

If you pay your underlings for exactly the number of hours you rebill them, you avoid all of the issues above: No employment taxes, no worries about 36-hour weeks or vacations or project gaps. It's simple time-arbitrage.

Terrific! Unfortunately, subcontractors **charge a lot by the hour**—a lot more than employees. Of course they do... exactly for the reasons above!

You haven't eliminated those effects, you've just moved them to someone else's profit-and-loss report. If you hired that same employee as a consultant she would charge you $60/hour instead of $30/hour and your profits are still the same.

Still, there's a lot less headache using consultants, so this might still be worthwhile. Simpler taxes (in America), no bargaining for vacation time, no prodding them to bill 45 hours this week to make up for last week. Even from a psychological standpoint this might be better: It's easier to give up $60 when you charge $100, but it causes some consternation to pay someone for a "day's work" when there's no work to do.

IS CONSULTING REALLY THAT BAD?

Consulting is a great way to earn a living and a smart way to self-finance a startup.

The trick is to avoid all these traps. For example, you know billing a full 40 hours per week is critical, so make it a habit to review hours weekly to make sure no one is consistently falling behind. As another example, set up incentives where employees get to share in the profits.

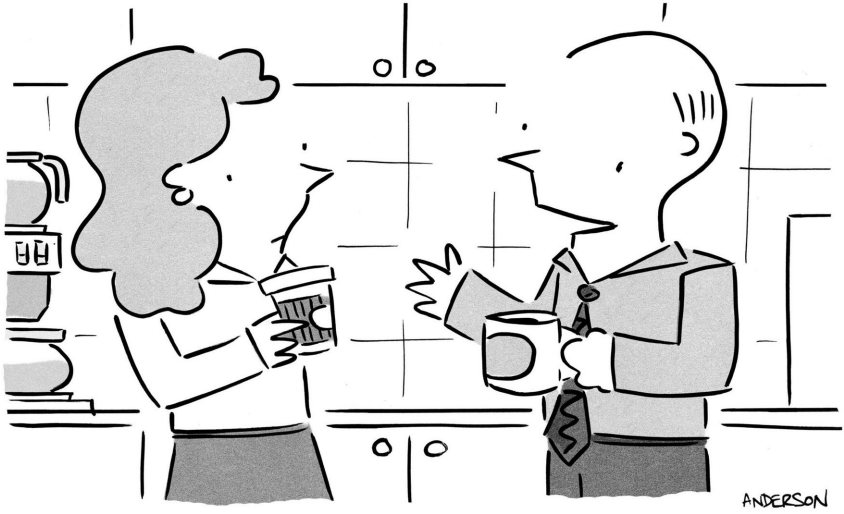
It's always hard. Most consulting companies don't make much profit, and it's one in a thousand that has the discipline to launch a successful product during off-hours. If you're going to make it happen, you yourself need to be serious, disciplined, and relentless.

But you *can* do it.

And if you do, you've just self-financed a startup, made a nice living, mitigated much of the risk of product-only startups, and built a great team in the process.

Chapter 47: JIT selection from independent streams: An alternative to the "big backlog" of work

PRIORITIZATION ≠ PLANNING · BIG BACKLOG
STREAMS · CROSSING



"Things have gotten a lot easier since I moved everything from my to-do list to my it-is-what-it-is list."

credit[744]

Scrum teaches us that the "single-threaded, ordered list of work" is the correct way to prioritize the work of the future. Innumerable articles—including one (p. 603) from this very site—tell us that a single, stack-ranked list is the simplest way to clarify priority.

I find, however, that this is unsatisfying at best, and fails to achieve sensible prioritization at worst. It also doesn't solve the political challenges of prioritization with stakeholders.[745]

The following system is simple, and has worked for me.

PRIORITIZATION ≠ WORK PLANNING

Prioritization means: *What is most important?*

Critically, it does **not** mean: *What will we work on next?*

Product Managers typically conflate these, trying to do both with a single process, and a single backlog. A traditional backlog efficiently encodes the answer to the second question, but isn't the right tool for answering the first question.

You might say, "But we should work on whatever is most important, right?" By unpacking the subtle answer, we'll arrive at a better way to prioritize, and a better way to plan work.

WHY "THE BIG BACKLOG OF WORK" ISN'T GOOD FOR PRIORITIZATION

Prioritization

If prioritization answers "What is most important," it begs another question: What do we mean by "important?"

- Important to calm our largest customer who is threatening to leave
- Important to catch up to our biggest competitor on a specific feature
- Important to unleash the sales and marketing teams onto a new market segment
- Important to satisfy our security policy
- Important to decrease the largest cause of support tickets
- Important to mitigate the largest cause of cancellations
- Important to refactor code so that engineers are happier and more productive
- Important to increase profitability
- Important to achieve our long-term strategy, even if there's no next-quarter revenue

Sorting a "Big List" of epics doesn't seem like the right way to tackle this complexity.

The first problem is that you're sorting things that are incomparable. Every bullet point above names something undeniably "important" but also belies a different motivation, with different impact, measured differently, so on what basis, with what metric, would you order The Big List?

The second problem is that the resulting decision is unsatisfying to stakeholders. Sales wants a few specific competitive features, and

doesn't have the context to understand how those are interleaved with requests from support or why the engineers need an entire month to "clean up a mess, which by the way, they made themselves?" The Big List doesn't yield satisfactory explanations.

Work-planning

The third problem is that The Big List conflates the idea of "prioritization"—what is most important—with "work-planning"—which is the order that specific work will be executed. While there's a correlation of course, often work-planning does not match our ideal prioritization because...

1. Hard deadlines—often externally-imposed—requiring work even if something else is nominally "more important." (e.g. zero-day security patch is not on our list of prioritized strategic work, but must be scheduled immediately. Or: Have to get specific functionality shipped for a live customer event).
2. We might not currently have capacity on the teams who are capable of working on a high-priority thing.
3. We might not currently have the skillsets to execute a high-priority thing.
4. Dependencies—i.e. A is very important; B is not. But A cannot be done unless B is done. Therefore, we must work on B first, even though A is more important.
5. Unknowns that have to be investigated before work can commence, or even be planned. (e.g. spikes or proofs-of-concept)
6. The amount of work, or amount of risk, of executing a high-priority thing might mean we shouldn't tackle it yet.
7. Getting success on some smaller things, while working on longer things, is better for throughput, better for morale, and is more agile.

8. Blockers—dependencies on other work, maybe even other teams, before we can make progress on a high-priority thing.
9. Starving—sometimes we've put off low-priority things for so long, they effectively become higher-priority.
10. Holidays and other seasonal things (e.g. kicking off a huge new initiative on December 15th is a bad idea, even if it's strategic).

The solution is to prioritize items in separate, actually-comparable streams, and to keep that prioritization effort cleanly separate from work-planning.

SEPARATE PRIORITIZATION STREAMS

> *"Don't cross the streams!"*
> *"Why?"*
> *"It... would be bad."*
>
> *—Egon Spengler, Ghostbusters*

The stakeholders

Folks in Sales may not know which items are most important for Support, but they sure as heck have an opinion on what would be best for Sales! It's fun to sit down with sales reps and sales leaders and ask them to generate and then stack-rank* the things they believe would

* Wait, that means a single, prioritized list! Yes. Because this is a single group, and we're asking them a single question from the bullet-list above (e.g. "what would be

be most impactful. Tell them they don't have to worry about other departments or other requirements—this time is just for them, and this list is just for them.

The ideas will pour out. Some ideas come from the last sale someone lost; that might be insightful or might be noise.** Some ideas are critical causes of lost sales; bonus points if you can tie them to specific lost-sales; that's motivating for Engineering as much as for Product. Some ideas are just a feeling—"I *know* I could sell the piss out of X." Don't dismiss those out of hand in your quest to be data-driven; multi-billion-dollar startups have been built atop experience-informed gut-instinct.

By forcing that group to stack-rank their ideas, they'll argue amongst themselves over which are most impactful. This creates empathy for what product managers have to do all the time—struggle with prioritization in the face of multiple goals and imperfect data. It also means they can't make everything "Priority #1," which in turn sets up a healthier future discussion: When, in two months, they complain that "you still haven't made feature X," you can point out that "X" was number seven on their list, so they themselves felt that it wasn't as important; you were just listening to their input. (Of course, everyone is allowed to change their mind; it's useful to run this process multiple times per year.)

You're not promising to do everything they want, in the order they want. That's work-planning, and also this is just one list; you're going to do this with other groups too. And this list doesn't include the most important voice of all—the voice of the customer. You *are* promising

---

best for increasing sales, whether because of increased win-rate or increased average transaction size"), we don't have the problem of comparing unlike things, and so the "single list" is a good forcing-function to generate a clear prioritization.

** One way to separate the short-term emotional reaction from the important trend is to observe what happens in subsequent prioritization sessions. When you see the group dramatically shifting the priority of items quarter-to-quarter, that indicates the group doesn't have a consistent sense of what would be most impactful. Surely some of the ideas are truly great; *this* team just isn't sure which ones those are. Sounds familiar—Product Managers struggle with this challenge all the time!

that you will take this list seriously, because you genuinely value their input. That's more than most Sales teams get out of Product Managers.

Now repeat for other groups.

Sales Engineering has a view unique from Sales, more technical and more in tune with technical customers, often the users rather than the buyers.

Support often knows the product better than anyone, and with the right tools and processes, has better data about which topics are causing the most problems or confusion. Remember that this isn't only about reducing work and costs for the Support team—although that's already a great result. It's about creating a better customer experience, because the customer who never needs to contact Support to get their work done, is a happier, more successful, more loyal customer.*

Engineering also needs their own list. There's always things to refactor, libraries to replace, spikes for new technology to try, replacement of a continuous-integration system, test automation we neglected before but desperately need now, systems that were adequate last year, but now we've scaled out of, and so on. Their stuff is about operational excellence and a happy, fewer-bugs, higher-productivity environment, not about growth or the customer. Therefore, it's easy for these things to never be prioritized, which in the long run creates resentment and grinds development velocity to a halt. But also, engineers—like everyone else—have more ideas than time, so they should be forced to stack-rank.

So... which of these lists are correct? There is truth in all, but none contain the complete picture, like the 2,500-year-old Buddhist allegory about the The Blind Men and The Elephant[747] (Figure 1).

It's the job of a Product Manager (p. 817) to consider all of these inputs and deduce the elephant. Keeping the inputs separate and stack-ranked helps the PM with this daunting endeavor.

---

* There are fascinating data[746] supporting the claim that "needing Support less" is correlated with "customer loyalty."

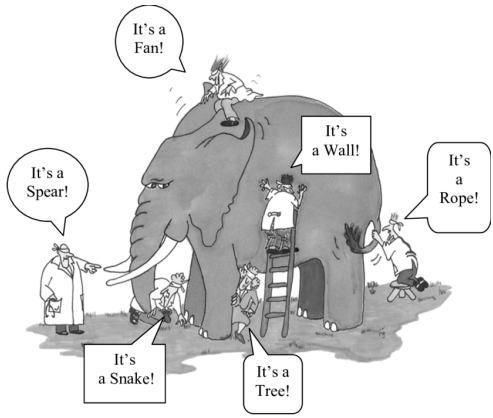


Figure 1: *Source: J. Himmelfarb (artist G. Renee Guzlas)*

The customer

These are internal groups; what about the customer, the star of the show! This is typically the purview of Product Managers and Designers and UX Researchers, as opposed to internal stakeholders. For simple products, perhaps there's only one prioritized list representing our best idea of what's important to the customer, but products with non-overlapping areas of functionality or that serve multiple personas might have one list per persona or functional space. So for example you might have separate lists for the needs of small versus large companies, or first-time users versus power-users.

Another common technique is to have a list per step in the customer journey, reflecting the fact that it's important for trial users to find success quickly, and also important for paying customers to continuously see value, and also important that the highest-revenue power-users to stay and grow. As with stakeholders, the motivation for the prioritizing items within each category are not comparable.

There is plenty of prior art on this topic. This is essentially what is happening with Story Mapping[748] or Opportunity Solution Trees[749]

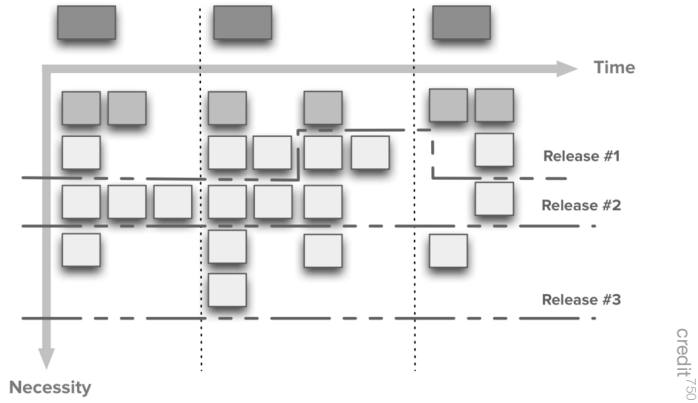


Figure 2

(Figure 2), where the customer-journey steps or the main opportunities each contain a list of potential work-items. Indeed, you could argue that story-mapping is just exactly the idea expounded here, extending the concept of "independent lists" from only customer-focused topics, to topics from Sales, Support, and Engineering.

Alternate types of list

If you don't like separating lists by function or by customer journey, here are other systems that some people like to use:

Kano

You could also create lists using the Kano Model.[751] Imagine one list each for Delight, Performance, and Must-Be. In general you don't want to starve any one of these categories, but it's difficult to prioritize items between categories. For example, it's easy to claim that everything in "Must-Be" has to be implemented first; after all it's called "Must-Be" for a reason! There's no point in bringing on new custom-

ers who won't be successful. That sounds reasonable, but it's simply not true that products should supply only basic requirements; indeed it's often the (unexpectedly) Delightful items that cause a customer to stay *despite* failings (p. 1271). Having separate lists, none of which should be completely starved, helps to combat this problem.

The "Four Lists"

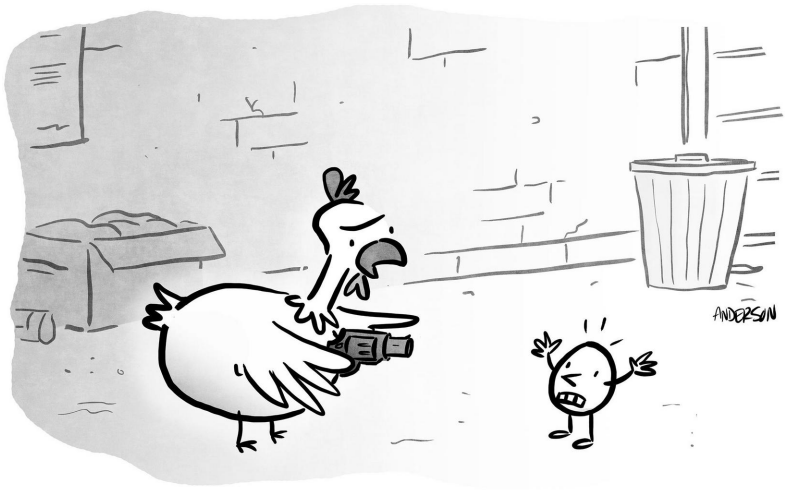
From Jyoti Bansal, summarized by Adam Thornhill:[752]

1. **Customer engagement.** Tuning into user feedback and desires.
2. **Sales intelligence.** Learning from the field to stay abreast of the competition.
3. **Technical debt.** The debt engineers must address to prevent performance issues.
4. **Vision for the future.** Where the product should head, expanding the addressable market.

In all cases, all of the lists are important, in the sense that you cannot ignore any one of them forever. But you don't have capacity to move all the needles at once, and the items within each list are different-in-kind, and should be prioritized separately.

CROSSING THE STREAMS FOR WORK-PLANNING

Eventually, human beings need to do work. It makes sense to select work from these various streams and—yes—place the work in an or-



"Maybe we'll never know who was first.
But I can tell you who's gonna be *last!*"

credit 753

dered backlog. When you're actually doing work, you want a clear answer to "what's next?" without additional meetings and debate.

By waiting until the last second to compare incomparable things, prioritization remains organized and clear. By waiting until the last second to shift your mindset from "most important" to "planning work," you maximize clarity in the process of selecting and scheduling work.

In software, when you wait until the last possible moment to do something, it's called JIT ("Just In Time"), hence the titular name of this technique.

You still have the unenviable job of selecting which items from which streams are going to happen next (or "soon," in the case of a roadmap). But that was always going to be the case—there's always more work than time, so there's always a selection process. This technique keeps that process more organized than it would have been, and it allows you to grapple with work-planning puzzles separately from

prioritization, which at least compartmentalizes the challenge. Divide-and-conquer is a good way to overcome complexity.

> *"We're obviously in no danger of arriving at consensus."*
>
> *—Warren Buffet*

You're selecting from the top ideas at the company, not from the 1000 ideas everyone collectively has. It's a lot easier to pick the best from 12 instead of from 1000, and in some sense you can't go wrong —on average all 12 are probably decent ideas. And stakeholders will be happy.

This last point is under-appreciated. This process is more explainable after-the-fact. Suppose it turns out Sales's first item is impossible to tackle right now, but you can knock out their second item because, by the way, it's also Support's number four and fits nicely into a multi-month customer journey improvement project you have. That's a great message to send back. Or, a message like: We knocked out two of your top items recently; we've been starving the engineering list so we're going to hit a few of those in the next few months, then resurface.

This process also helps keep yourself honest. You can easily report on:

- How much of each kind of work are we doing?
- What is the age of the top three items on each list? (i.e. Are we starving something?)
- How much of this sprint is addressing someone's "top 3" thing?
- How much work is related to internal projects versus customer-facing?

It also leads to better ideas:

- Are there trends that everyone is seeing across all functions in the business, that therefore should inform our strategy?
- Can we think of new product ideas that hit multiple top-three items simultaneously, and thus are especially impactful?

Don't forget to schedule rocks, then pebbles, then sand (p. 221). That's an even more primary principle for work-planning. Separate prioritization streams help identify which rocks and pebbles should be scheduled in the first place.

> *"When we differ, Charlie usually ends the conversation by saying: "Warren, think it over and you'll agree with me because you're smart and I'm right."*
>
> *–Warren Buffet on Charlie Munger*

Chapter 48:
Ruthless prioritization while the dog pees on the floor

MANDATE · 10X / 0.1X · FALLOUT

"Remember that thing we thought was a thing?
It's not a thing. But here's the thing, that *other* thing?
Totally a thing!"

THE PRIORITIZATION MANDATE

Time is a zero-sum resource: An hour on one thing necessarily means *not* spending an hour on the entire universe of alternative things. Every minute is a choice. Every choice is a trade-off.

Time is a hard limit. We can maximize *productive* time through better habits: good sleep, sensible diet, reasonable exercise, restorative rest, reading books instead of doom-scrolling Twitter, sleeker task-management, fewer meetings. But even here we're prioritizing within 24 hours per day, choosing how we labor and rest to maximize our effectiveness. There isn't enough time for more than two big things (p. 901) in your life.

> *"All we have to decide is what to do with the time that is given us."*
>
> *—Gandalf the Wizard*

We need more time. Not 30% more, but 3000% more.

Every company proves that statement with a repository of data called "the issue tracker". You accumulate thousands of valid items over the years: Little ideas that would be nice to do; big features that would create differentiation (p. 949); bugs that paying customers actually experience; design tweaks (p. 853) that make the team proud of their craft. You will never complete all these; completing even 10% would be a miracle, the ratio worsening every week as more items

are added than removed. The better you listen to customers and the more creative your teams, the worse the ratio becomes. You hire more people to get more done, but new talent have new ideas, and the ratio worsens again. Inspiring, and maddening.

So, time is a fixed constraint that limits us to 1%-10% of what we "need" to execute. Unintuitively, the fraction diminishes with scale; we will never defeat it.

The inescapable conclusion is the trite statement that "We must prioritize"—intelligently determining which precious few things we will actually do.

In fact we can't help but prioritize, even if mindlessly. Since we can only do one thing at a time, whatever we're doing now is definitionally our "highest priority." Reading this sentence is currently your highest priority. While "prioritizing" doom-scrolling is obviously faulty decision-making, there's the more insidious cases of prioritizing things that are fun but only somewhat useful, or of prioritizing things due to necessity but not importance (e.g. paying taxes on time). Indeed, a common complaint of prioritization is that we're only doing things that happen to have deadlines, instead of things that matter. "Urgent, but not important" tends to win over "Important, but not urgent," if we're not paying attention.

Prioritization is a choice, and more often than we'd like to admit, the choice was thoughtless. We've all lost a few hours to doom-scrolling or YouTube or TV (if you were born before 1990). Maybe that activity was useful for "restorative rest." Maybe it was just a poor choice. Maybe we should give ourselves grace. Maybe.

Prior art

There are, of course, myriad prioritization frameworks. On this site alone I've detailed many of my own:

- Fermi ROI (p. 171): Replacing rubrics, especially "ROI"-style

- Binstack (p. 603): Making significant choices with incomparable dimensions
- Adjacency Matrix (p. 793): How to expand an existing product
- Investment Criteria (p. 867): When to invest significant time and money
- Rocks, Pebbles, Sand (p. 221): Analyzing and prioritizing three *sizes* of work
- Satisficing vs Maximizing (p. 887): Prioritizing some things (p. 645) as "good enough," others as "never good enough"
- Leverage Points (p. 1125): Where incremental change yields large results
- Cleaving (p. 1303): Separating upside from downside, treating each differently
- Fairytale Quarterly Planning (p. 1065): Prioritizing work against strategic objectives and the obstacles that are preventing us from winning
- JIT streams (p. 711): Handling multiple, incomparable inputs, separating prioritization from work-planning

I don't like anything with a rubric or a computed score (this is why (p. 171)). I don't like anything with a "confidence level" (because you can't discuss it accurately (p. 997)). I don't like anything that is built to produce symmetry (p. 465) for consultants' slides rather than reflecting the messiness of the real world. I don't like anything that purports to compare incomparable things (e.g. scoring "more growth" with the same made-up number as we score "don't run out of money" or "make employees happy").

Beyond the built-for-purpose frameworks above, there's a simple overarching framework that applies to every type of prioritization, foisted upon us through the observation that we have time for less than 10% of what we'd like to accomplish.

Here is that framework.

10X / 0.1X PRIORITIZATION

10x tasks

Despite how precious time is, some tasks are so valuable, so impactful, that the return on your investment is an order of magnitude more than what you put in, even if you valued your time at (say) $1000/hour (p. 1413).

You must seek out these "10x things"[755] that can transform the company. Examples:

- The few features that win the majority of sales, whether through pure delight (p. 275) or because of a combination of utility and uniqueness among the competition. Rule of thumb: Impactful features are actively used by at least 40% of customers, or are a critical reason-to-buy or reason-to-stay for at least 15%.
- Finding the ideal marketing positioning (p. 165) and wording (p. 627) where advertisements convert 2x higher and people landing on the home page buy 2x more often. (Not 10% higher—you can't actually measure that (p. 913).)
- Finding the pricing model (p. 515) that maximizes profitable growth while maintaining fairness for customers.
- Taking the time to correctly identify the next strategic objective or biggest obstacle (p. 1065), so everyone can prioritize their own time towards this most-important thing; working on the wrong thing is a 100% waste of time, even if efficiently- and perfectly-executed.
- Hiring the next critical employee who dramatically increases the company's throughput and work-quality and decision-making-quality, while adding a skill-set that was previously missing, a skill-set needed to overcome the current obstacle or the next strategic objective.

- Applying energy to a Leverage Point (p. 1125)—an area where even small changes have a large impact on growth or profitability.
- Addressing the single biggest drag on growth (p. 1191).
- Deciding how to expand the business into the next adjacency (p. 793).

If you don't know what one or two 10x tasks you should be working on, then *identifying that is your highest priority*. Otherwise are certainly not prioritizing properly; the entire company is misusing their time. This is my method (p. 1065) for determining what those things are.

If you have too many 10x choices, the Binstack prioritization framework (p. 603) is designed for this type of decision, where the inputs are incomparable, and you want to maximal impact.

It's rare, however, to actually have too many 10x possibilities. If you think you do, it's likely that you are being too generous in declaring things "10x". They need to literally "10x" a key metric, or be the difference between survival or bankruptcy. Their upside must be so massive, that even when it inevitably takes 2x longer to implement and is $\frac{1}{2}$ as impactful as you thought, it was still worth it. To take that 4x hit and still have a great outcome requires starting at 10x.

To further refine your thinking, treats these are *investments*—i.e. applying significant time, expecting an out-sized return, but with uncertainty. Here is a guide for making good investments (p. 867).

0.1x tasks

Most activities are worth far less than $1000/hour, even if they are mandatory. These are the "0.1x tasks." You must minimize these through several strategies:

- Eliminate them completely by structuring your life, product, target-customer-selection (p. 317), or company strategy (p. 489)

to avoid them. Here is a guide (p. 569) for inventing ways to avoid them.

- Delegate (p. 981), even if the result is worse than what you'd do yourself (e.g. grocery delivery being both more expensive, and not picking the same apples you would have picked). It's not worth the time to do everything (your definition of) "perfect." You need to be in command instead of in control (p. 413).
- Batch or automate,[756] accepting minor penalties (e.g. paying bills only once per month and risking occasional late fees, or batching security patches, since you can't ignore them forever but each one is unlikely to be exploited in the next few weeks).
- Archive loose tickets that are older than 100 days. Because of the rule that more than 90% of our ideas will never be done, realize that these are already among that 90%; deleting them will help you prioritize the remainder. If something is truly important, it will come up again in future. (Hence "archive", not "delete".)

It's tempting to assume that small, easy tasks are automatically 0.1x tasks, but that is a fallacy of conflating "impact" with "effort". Easy things can have a large impact (making them 10x almost definitionally), and complex things can have no impact on revenue or employee happiness.

In some regimes, many small tasks add up to a 10x impact; I call this "life by a thousand sparks".* A common example is "great design". While amazing design is not required for success (p. 853), there are many examples of products winning primarily because of beloved design. Great design is not "one thing," however. There are macro-scale architectural decisions, but also it emerges from thousand details: subtle color choices, pixel-perfect layout, font and word selection, aspect ratios, play of white space, satisfyingly snappy interactivity, a design system for comprehensive consistency, fixing every last bug that is aesthetic rather than functional, and harmony between the

* A facetious inverse of "death by a thousand cuts".

credit [757]

"I *am* multi-tasking! I've done this report three or four times already!"

website, the product, the emails, and the material for marketing, sales, and support. These myriad tasks should not be dismissed as individually 0.1x, *if* they are specifically in service to a 10x concept that is also the primary way you win customers in a competitive market.

What if there were no 1x tasks?

Prioritization isn't a dichotomy, but I encourage you to act is if it were, forcing yourself to make clearer decisions. Idea-abundance is beautiful, but we must be ruthless, final, and precious with our time.

You will inevitably label things as "10x" which are really 1x. You will spend too much time on 0.1x things; after all, there are so many. Forcing this binary choice is reductive, but helps us be ruthless.

Of course in reality there is a spectrum of ideas, sizes, impacts, risks, and confidence. Our ability to measure any of that even *post facto* (p. 1253) is laughably poor, and our ability to predict any of them (p. 1141) is even more pitiful. If you insist on the existence of 1x

tasks,* use the Rocks, Pebbles, Sand Framework (p. 221) which explains how to segment by size, how to prioritize each size differently, and how to resolve the common conflicts that arise as you schedule work in the messy real world.

DEALING WITH THE FALLOUT: DOGS PEEING ON THE FLOOR

In a well-lit living room, a man sits in a chair, reading a book. A dog nearby is peeing on the floor. The man doesn't react. You're watching from a window, concluding that this man must be ignorant, crazy, or at least a poor decision-maker. Put down the book and take that dog for a walk, idiot!

Except, you don't know the full story.

In one hour, the man has the most important meeting of his life. His performance in this meeting will dictate the next ten years of his career. Everything he needs to know to be successful in this meeting, is in that book. Yes the dog should have been taken for a walk, but the penalty of having to clean up the pee is worth it, because the call is *that* important.

The decision is rational. Yet the observer sees only irrationality. This is caused by two things:

- The observer doesn't know the complete story.
- The observer has trouble accepting that something bad/stupid can nevertheless be the right decision, because the alternative is even worse.

---

* Another exponent of 10x / 1x / 0.1x is the great Product Management (p. 817) teacher Shreyas Doshi[758] with his excellent LNO framework.[759]

This inevitably happens at your company when you ruthlessly prioritize. Because you're so focused on the *most* important thing (p. 1119), other things lay fallow. Important things. Fires burning that you're intentionally ignoring because although they *are* fires, they still aren't as important as the Most Important Thing. Dogs peeing on the floor. On purpose.

But others see the fires burning, the dogs peeing, for months, for years, and then lose faith in leadership. They complain—understandably, and accurately—and lose morale. Worse, they start believing that The Deciders must be crazy, just as the window-peeker believed. Loss of trust and respect leads to talent leaving, which leads to the death of an organization.

"They never listen to my ideas," the most prolific idea-creators complain. They're right, too; after all, mathematically you could never implement more than a few percent of those ideas. Not because you're "not listening," but because of inexorable math. You have to focus on the peeing dogs, and you can't do most other things.

The way you combat this natural progression is to address the two bullets above. You have to transmit the complete story, not just of the few priorities, but why they are the top ones. You have to acknowledge the twenty things other that also deserve attention, explaining why we're intentionally letting those fires burn, those dogs pee, because the top priorities are even more important. This is why the "not doing" list is such an important part of the plan.

And you can retell the story of the dog peeing on the floor, acknowledging that it feels bad watching the dogs pee, especially when you know how to prevent it. I don't remember the origin of this particular parable, but I've retold it many times at WP Engine, and it sticks. It works.

This is what prioritization actually looks like. The full picture, not just the social media admonitions that you "have to focus! (p. 1119)".

The full story requires constant, repetitive communication. Because it *is* bad to see the pee; we can rationalize it only when we see that the pee buys us time to do the most important things—the things that leverage our precious, woefully limited time, for out-sized results.

No one said it would be easy. Good luck.

Chapter 49:
When you want to quit because it's just not worth it

I've been there too. It sucks.

Most startups fail "only" because (p. 379) the founders stop working on them, and often it's because it's emotionally draining. I don't care who you are or how strong your ego is, you *will* have these moments—perhaps a continuous stream of moments—when you can't take it anymore.

I cannot remember the number of times I was so overwhelmed at Smart Bear that I almost threw in the towel. Close the bank accounts, close the doors, turn off the website, bounce the email, and just...

stop.

Sounds dramatic, but it's no exaggeration. You'll hit these walls too. I know you will because this is what people admit in little rooms

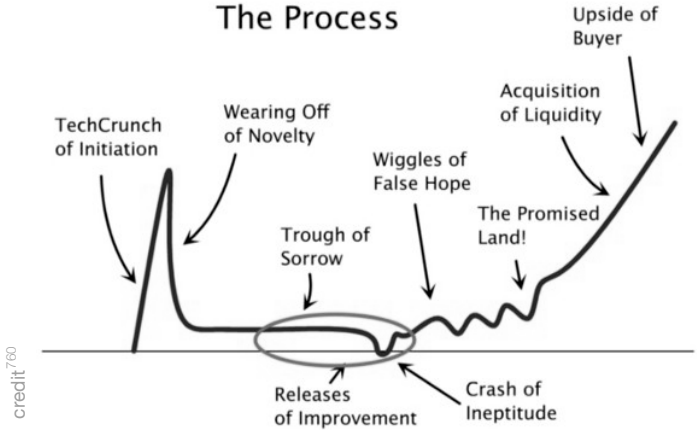


Figure 1

with other people who truly understand and won't name them publicly, and also there's a steady stream of people saying it out loud on Twitter, many times per day.

Maybe a little commiseration will help you get through it.

You expect these moments to happen at the *beginning* of startup life—when you're least confident, have the worst product, and the least knowledge about your customers and the market. Paul Graham and Trevor Blackwell captured it pretty well (Figure 1).

The pain is not limited to the beginning of the venture. It's still there years later, despite real revenue, profitability, customers arriving every day, and a great team.

Since *that* is *not* obvious, I'd like to share a personal story.

Four years into Smart Bear I had several employees getting paid decently (which at a bootstrapped startup is hard to do!), a product that people were buying, and we were doing around a-million-a-year. Life was good!

I was negotiating my first true "Enterprise sale." This was going to be our biggest order to date—something like $200,000. (Yes, one-time revenue. It was a simpler time.) Actually "negotiating" is the wrong

word because I don't believe in price negotiation (p. 895), even with enterprise sales (an area that most people claim *must* include automatic discounting.)

The person with whom I was negotiating wasn't the end user, nor the boss, nor the boss's boss, nor anyone in that chain of command. See, big companies have entire departments devoted to dealing with vendors like you and me, and when it comes to negotiating, these departments **harbor terrorists** with titles like "Procurement Manager" or "Strategic Sourcing Manager."

I say "terrorists" because they use fear tactics to get their way, yet they have no power other than fear. Imagine the worst stereotype of a salesman, except instead of selling you something, their job is beat a discount out of you.

Now to be fair, many vendors do take advantage of large companies—overcharging (because "They can afford it!"), or promising one thing to the users and sneaking something else into the invoice.

But mostly it's because of the traditional enterprise sales dance, reminiscent of the lumbering mating dance of the great blue whale. The vendor asks for too much money; the client is astonished at the price. Both calmly explain that this is a deal-breaker. Then the vendor capitulates 30% but only if the client signs a three year maintenance contract (which they wanted anyway). The deal is struck.

This tradition continues because of perverse, wasteful incentives. The vendor's salesman likes this because sometimes she gets away with a high price which pads her commission check. The Procurement Manager likes this because he can show his superiors how much money he's "saved" the company.

So big companies need a Defender of Evil Vendors, I get that. But that's not enough for these guys; it feels to me like an attack, not a parry.

This is how the conversation went:

> PM: What kind of discount are you offering?
> Me: We don't discount; instead we put our pricing on our web-

> site so there's no misunderstanding.
> PM: Well I'm going to need some kind of discount. How about 30%?
> Me: As it says on our website, we don't discount.
> PM: But I'm buying 400 seats!
> Me: Yes, and we already provide a nice discount for bulk orders, which is already included on the invoice and documented on the website.
> PM: You don't understand, I *always* get a discount. I've done business with 47 other vendors and *all* of them give me at least 20% off.
> Me: There's always a first!

So far it's actually OK–*I'm* the one refusing to plod through the mating ceremony, wanting to skip right to the wedding night. I expect push-back.

Here's where it gets nasty. I remember sitting there on the phone getting lambasted for my intolerable ignorance about the Way It Works. I was told, and I quote, "**You have no business selling anything to anyone.**" My obstinate ignorance is a deal-breaker for the entire sale because of what it implies about my company in general–after all if I don't even understand the purchasing process there's no chance in hell my software's going to work! Furthermore, despite my ignorance, I'm unwilling to listen to the rules, unwilling to learn, which means there's no hope for me.

I'll never forget how this ended:

> PM: OK that's it, you give me no choice. I absolutely cannot approve this deal, and furthermore I'm recommending that **we never work with your company in any capacity**. At this point, even if you gave me a discount I would still reject it.

Here's where I'm supposed to unleash my intellectual fortitude. I won't capitulate, will I? I won't let this guy insult and bully me, will I? C'mon, I'm the strong-willed confident entrepreneur with the stoic well-argued voice of reason, whereas he's the sleezeball with the tedi-

ous day-job—surely I'll laugh as his words roll off me like water off an oiled duck's back.

Just the opposite. I felt like throwing up. He's right, who do I think I am? I'm a baby geek playing in the adult's house and I don't know what the hell I'm doing. I have these naïve ideas about how the world *should* work and how people *should* treat one another, and it's all just incredibly silly and ignorant. And it *shows*. And now this guy is going back to those other folks at the company whom I actually like, and worked hard to earn their trust, as they laugh together over can-you-believe-how-dumb-he-is and we-can't-possibly-do-business-with-them.

It's over. They've seen through me (p. 457). It's just a matter of time before others do too. That's the end of deals like this.

Why am I doing this anyway? This is supposed to be fun and fulfilling but at this moment as we say in Texas I feel like ten tons of shit in a two-ton bag. What I like is writing code—why am I even trying to play this sales game? Why not just go get a job where I only worry about whether or not I can write code—because I sure as hell can do that—and let the natural salesmen do all this crap?

Is the money worth it? What money, we're still bootstrapping and I still don't get a regular salary. Is the *promise* of money worth it? Worth these feelings of inadequacy?

After days (yep, *days*) of fretting like this, it converted from despair to anger. Who the hell is this guy? Some asshole who isn't good enough with money to be an accountant, not even slimy enough to sell cars, this guy whose only skill is to be a jerk, some guy who has never had to make payroll or take a risk or put himself out there, this schmuck is going to tell me *I'm* the one who isn't good enough, *I'm* the one who has no business selling software?

Worst of all, I'm *letting* him make me feel like a pile of shit!

Well if you're waiting for the big moment where intellectual reasoning finally defeats weak, irrational emotions, I'm sorry to disappoint you, because **that moment never came**. I know it's dumb and

illogical, but there it is. It's trivial and baseless but I still carry that experience in a corner of my thoughts. That's how emotions work.

By the way, this guy turned out to be totally full of shit. He had, in fact, no power to stop the deal. When I finally got my main buying contact from that company on a conference call with the PM, the conversation was this short, and as close to word-for-word as I can recall:

> My Guy: So, what's holding up procurement's approval?
> PM: Nothing, just some paperwork, we'll have it done by Friday.

All of that angst for nothing. Son of a bitch!

Years later I was on-site at this company and I finally met this guy face to face.

I *still* felt small.

Want to say I'm weak? Or he's strong?

Who cares, the point is: Getting through this slog of a thing that's a startup—or anything difficult and worthwhile—doesn't require that you're always confident or stoic or smart or right or wise. You don't need to match the emotional stability you see from the big bloggers. (Which is mostly a façade[761] anyway.)

It's not even about "overcoming." Maybe you don't overcome, you just get through.

It's about sticking through the tough parts, whatever your foibles and weaknesses.

***Living through* it, not *beating* it.** I never have, to this day, "beaten" that PM, not emotionally, not if I'm being truly honest.

Welcome to your crucible.[762]

I'm not saying tenacity is *all* it takes. Just that without it, you'll stop.

It's so easy to stop.

There's so many reasons to stop.

And that—stopping—is how most little startups *actually* fail.

P.S. Since writing this article in 2011, I was able to avoid burn-out and tame these thoughts with a framework that I detail in this article from 2022 (p. 399).

Chapter 50:
When should a decision be fast, or slow?

"I haven't yet decided if I'm staunchly in favor, or staunchly opposed. But rest assured, however I vote, it will be staunchly."

We all know that startups should make decisions quickly.[763] Fast decisions leads to rapid action, which accelerates the loop of production and feedback, which is how you outpace and out-learn a competitor, even one that already has a lead.

But some decisions should not be made in haste, like a key executive hire (p. 1303), or pricing strategy (p. 515), or whether to raise money, or whether to invest millions of dollars in a new product line, or whether to enter a new market.

How do you know when your current decision should be made slowly: contemplative, collaborative, deliberate, data-driven, even agonizing?

The following scorecard will help you know whether it is wiser to go slowly:

You'll make a much smarter decision later
Today we know the least that we're ever going to know. Often, waiting six months doesn't automatically give us more information, but sometimes it does. If we haven't launched a product yet, we're in no position to decide what the next few features should be; we should launch and earn experience through customer interactions. We'll then be much smarter about answering that question. So, don't waste time even thinking about it now, and certainly don't make any firm decisions, when you know you'll make a better one later.

Can't undo
This is the classic one-or-two-way door[764] delineation. If you can't easily undo the decision, it's worth investing more effort into analyzing the likelihood of the upsides and risks.

Huge effort
Some things take less time to implement than to estimate or to debate (i.e. Sand (p. 221)). It might take two engineers one week to implement something, but a few debates and a time-boxed research project could involve an entire team for a week. This is a reason why small teams without process can produce results faster than larger teams with process. If the effort to implement the decision is smaller than the effort to make a decision, just knock it out. But if you're deciding on a path that

could take six months to measure results from, taking time up front to research is wise.

No compelling event
If the status quo isn't bad, there might not be a reason why a decision should be made quickly, or at all. Without time-pressure, it's justifiable to spend more time on the decision. Conversely, time-pressure means the more time you spend deciding, the less time you have for implementation and unanticipated problems, so you're actually adding risk by dragging out the decision.

Not accustomed to making these kinds of decisions
Online marketing teams are accustomed to throwing creative things at the wall, because that's the day-to-day reality of their job. Because they're good at it, they don't waste time hang-wringing over whether or not to try an advertising campaign on the latest social media platform, or which headlines to try; they just do it. Conversely, most organizations have no experience with major decisions like pricing changes or acquisitions, and most founders have no idea how to hire a great executive, or how to decide whether to invest millions of dollars in a new product line

"We're taking a wait and see and then do nothing approach."

as opposed to "just throwing something out there and iterating" as was the correct path at the start of the company. When the organization has never made this type of decision before, the decision is at great risk, and being more deliberate with research, data, debate, or even outside advice (p. 449), is wise.

Don't know how to evaluate the options
Even after generating the choices, does the team understand how best to analyze them? If the company's strategy is clear and detailed, if relevant data is at hand, if it's clear what your goals are (p. 887), if the deciding team has confidence, then the decision could be easy and fast; if these things are absent, perhaps more deliberation is needed to clarify those things. (Another technique is to use Fermi Estimation (p. 171) to simplify the evaluation, or Binstack (p. 603) to find the one best option among many.)

Can't measure incremental success
After the decision is made and implementation begins, can you objectively tell whether things are going well? If yes, it is easy (p. 159) to course-correct, or even change the decision, in the presence of reality. But if progress will be invisible or subjective, such that you will sink person-years of time into the implementation before knowing how things are going, it's worth spending more effort gaining confidence in the path you've selected.

Imperfect information
Buying a house is nerve-racking, mostly because it is likely the most expensive and difficult-to-undo purchase of your life, but also because you know so little about the goods. What does the seller know but isn't telling you? What will you not discover until you've moved in, or a year later? Often it is impossible to get the data or research you need to make an objective decision. When this is the case, it is sometimes wise to spend extra time gathering whatever information you can, maybe investing in reports or experts (which is what you do with a house). Or you could look at it the opposite way: If it's impossible to get objective data informing the decision, then don't spend lots of time debating subjective points; just make the decision from experience and even gut-check, because we just said that's all you have to go on anyway, and experts are often wrong (p. 193).

Decision requires multiple teams who haven't worked together before At WP Engine we're extremely collaborative across teams. The benefit is that we work together for a common goal, taking care of the needs of support, sales, marketing, engineering, product, and finance, rather than solving for one department's goals at the expense of another. But this also can make decisions more difficult, because finding a good solution is complex, often requiring compromise or creativity which requires time to be realized. This effect is amplified if the teams (or team members) haven't worked together before, and thus have less rapport, common language, and common experience. In that case, give the decision more time to breathe and develop, because really you're giving *people* the time to build relationships and discover great solutions, and that in itself is a benefit to them and your organizational intelligence, which is a long-term benefit worth investing in.

Actually this isn't a scorecard, because important decisions aren't a Cosmo Quiz. Don't use this as a rubric; don't score it 1-5 and add it up with a spreadsheet.

"Ma'am, there's really nothing I can do until they've been missing for at least 24 hours. In my experience, the best thing to do is just leave it alone and they'll come home, wagging their tails behind them."

credit[765]

Rather, honestly answer these questions, and by the time you're through, you'll have a good sense of whether a light touch, quick decision is fine (which is the default answer), or whether you should take more time.

And, depending on which pieces are problematic, you'll have a guide for what needs to be done next.

For example, if "Can't undo" is a big problem, can you rethink the solution so that it *can* be undone, maybe by applying it only in one segment, or investing more time in planning or preparation, or creating a disaster recovery plan, or splitting up the decision so that part of it is undoable?

Or if "No compelling event" is a problem, maybe the best answer is to "not decide," i.e. don't spend time on this right now, since you don't have to. Some people will be disappointed in the lack of a decision, but it's better to honestly state that "we can't figure out the answer right now" than to make a rash decision that does more harm than good, or to invest time in a decision that doesn't need to be made, at the expense of work that does need to be done.

I hope this helps you make the right decisions, in the right way.

Chapter 51:
Distinguishing constructive criticism from bad business advice

FRANKS & GERRYS · DIETING
FINDING YOUR TRUTH

FRANKS & GERRYS

Haiku:
With their eyes of ice
high-powered executives
know better than you.

I was starry-eyed when Frank showed up 18 months after the birth of my company, Smart Bear. Picture it: Frank was a silver-haired ex-VP-of-Sales for a big, successful company. An IPO had made him comfortably wealthy, and now he wanted to hook up with a promising new startup. With his résumé and enough money that he never had to work again, he could be choosy.

And he chose little ol' me!

And boy did I (think I) needed him. "I'm just a software developer, I don't know anything about sales" I explained to Gerry, who had already unknowingly cemented himself as my mentor. "I'm just a lowly engineer, ignorant of the mystical voodoo of six-figure purchase orders. I don't even play golf! *Frank has been there*."

Gerry tried to explain. "There's no magic to it. I've sold companies for $100 million without stepping foot on a golf course. You have a good product, you have some customers, why do you think you need to change? What would happen if you just kept going?"

Sage advice! So of course I didn't listen.

Frank and I headed down the road to partnership. Frank had lots of ideas that "we have to implement, otherwise no big company will give us money." For example, we had to change our name. "Smart Bear" sounded to Frank like a dorky one-man shareware site. (Ahem (p. 1559)...) We needed something stoic and corporate. His suggestion was "Software Test and Deployment Systems, Inc," or STDS* for

short. "Big companies like acronyms," he explained, as if Big Company were an entity capable of having an opinion.

There was more: We needed patents ("to keep competitors at bay"), we needed a sales team ("software doesn't sell itself"), and we needed to lighten up ("I like to knock off early on Fridays"). In fact, very little of what I was doing, was right.

Fortunately Gerry finally punched through Frank's dazzling fog. "Let me get this straight," he rightly admonished me for the fourth time, "this guy wants 50% of the company, he wants a salary while you get none, he's going to work 35 hours a week while you work weekends, he's not investing money into the company, and his big idea about how to get more customers is to change your name to venereal disease? Does this sound like a 50/50 partner? Or even someone you want involved at all?"

Gerry's criticism of Frank was harsh—as harsh as Frank's criticism of me—but Gerry was right and luckily I (finally!) heeded him.

Two weeks later I landed a $50,000 deal with a Big Company. (It was Intuit.) **A few years later we were doing millions in revenue with Big Companies, still called Smart Bear, still no patents, still no sales guys.** Who knew? (Gerry knew.)**

You have to understand though, saying no to Frank was hard. He had the expertise; I didn't. I was convinced that he was right and I was wrong. If it wasn't for Gerry's guidance, I would have gone through with it.

---

* It apparently didn't occur to him that this acronym is already taken. At least we could have leveraged viral marketing! (I could have said "used" but Frank would have said "leveraged".)

** Editor's Note: I later sold Smart Bear (p. 45), and now it's worth[766] over $2,000,000,000. And it's still called Smart Bear.

So how do you separate the good counsel from the bad—the Gerrys from the Franks? Both sounded like practical advice and criticism, both were experienced, both were strong-willed, and both truly believed in what he was saying.

DIETING

The answer can be found in the brutal world of dieting.

One thing we've learned from the diet crazes since the 1980s is that every single thing has been alternately touted as healthy or poison. No-fat, carb-heavy. Scratch that, no-carb and fat doesn't matter. Scratch that, it's only about low-cal. Scratch that, whole-30 and don't track calories. Scratch that, fast for sixteen hours a day and do anything for the other eight.

Ask any person and they're equally variable: Which thing worked for them, or didn't, or worked for a while but it wasn't sustainable.

So there is no objective answer to the question: What is the best diet?

The only question is: What works *for me*?

Start-up advice is the same. For every example that incontrovertibly proves "X is right," there's another equally compelling story of success where the mantra was "X is wrong." The company that won because prices were low (p. 437) and the company that won because prices were high (p. 1223). The founder who built the company on top of a thriving Twitter following[767] with no advertising and the company where the founder ran ads[768] and isn't on social media at all. The

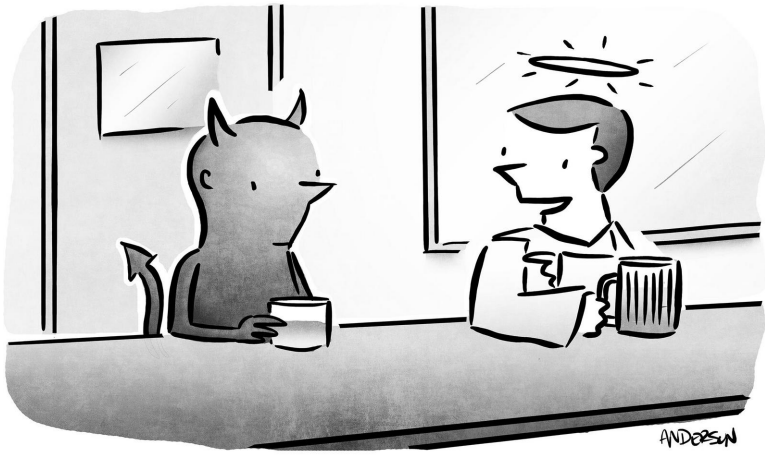
company that started because "I scratched my own itch (p. 533)" and the company that started because "I researched and located an underserved niche in the market (p. 71)." The company where the founder won because she was an expert (p. 1427) in that field and the company where the founder won because she adopted a new perspective (p. 891) in that field.

So how do we answer the question: **Which advice is right *for me*?**

In diets, half the answer is physiological—how your body reacts. The analog in startups is: What is right for this company, in this market (p. 71), with these competitors (p. 1061), with these customers (p. 317), at this price-point (p. 515), with this business model,[769] with this team, with our strengths (p. 543), with our goals, with our financing (p. 353), with our culture (p. 861). Usually the answer is different from what made sense for Steve Jobs or Bill Gates or Mark Zuckerberg or Jason Fried or Pieter Levels, even though those are the stories we constantly hear. Indeed, all of those are outliers even among their contemporaries; no one should attempt to follow their unique paths.

In diets, the other half of the answer is sustainability—can you keep doing this for a long time? Diets work only with a lifestyle change, not when it takes fresh willpower every single day. It's fine to say "social media is the key (p. 1027) to your marketing success", but if you think Twitter is insipid[770] and Facebook is unserious and Instagram is fake, will you really be successful if you force yourself to post there? Or should you follow the path of most companies, whose marketing is not primarily powered by social media? It's fine to say user interface design is critical, but thousands of successful companies have crappy design (p. 853), so if you're not a designer and don't care to invest in one, you should be asking what makes companies successful *despite* poor design (p. 1239).

How do you determine those personal dimensions ("Who am I?") or the corporate ones (the list above)? This takes real work; this article explains my system for finding both (p. 569).



credit[771]

"Remember when we used to shrink down, sit on people's shoulders, and give them conflicting advice? What a hoot!"

FINDING YOUR TRUTH

Beyond the tools in that article, the following have helped me discern the Franks from the Gerrys:

Insight, or an excuse?
Ask yourself: Do I like this advice because it's justifying some behavior that deep down I know is actually wrong? Or do I like it because it's clearly articulating a ground truth? For example, if you inherently dislike social media, of course you can find advice telling you how social media isn't important, but did you enjoy that because it feels correct deep down that social media is a farce, or did you enjoy that because it's an excuse to avoid the truth (p. 657) that social media is critical in your chosen market. If you proactively ask yourself—even though these are feelings—the answer will often be clear.

Context
Advice is valid only within a certain context, yet the boundary of that context is often unknown to both advisor and advisee. The rule of thumb is: Advisors give advice to themselves. Meaning: Within the boundaries of their personal goals, their world-view, the markets and products and customers and competitors that they've experienced, they can probably provide some great advice. Outside of that, who knows? So ask yourself: Does the person behind the advice match me on all those dimensions? If so, this could be relevant wisdom, even if the message is hard to hear. If not, you can either ignore it completely ("Focus!" (p. 1119)) or sift it through a substantial mental filter.

Questions, not answers
The best advice doesn't come as a barrage of commands but rather from a series of questions, asked by a devil's advocate. Rarely will an advisor know more than you do about your domain of expertise but that doesn't mean an outside voice is useless. Pointed questions force you to defend your choices. A healthy debate challenges your assumptions without implying they're false. New ideas are batted around as a brainstorm rather than handed down as gospel. Even Roger Federer has a coach, not because the coach is "better at tennis" or giving him commands, but because he is a mirror and a provocateur. In my case, Frank presented his view as a series of statements; axioms, even. Gerry couched his perspective as pointed questions that required either rebuttal or agreement. In fact, playing devil's advocate is a great exercise to do periodically. Find an intelligent foe, take her to lunch, and follow Scott Berkun's advice about Rude Q&A (p. 1533).

Some learning starts with revulsion
Pay attention to advice where you have an immediate, automatic revulsion. Often this feeling is correct, because you're want advice that matches your context, and revulsion indicates a mismatch. But, if you are wrong about something, this is what it feels like to discover that. This is an opportunity; don't waste it!

"This is how it's done"
"This is how it's done" is almost never a good reason. If the sole basis for the advice is "tradition," it's just momentum and you might be right to buck the trend. Example: Smart Bear posts its prices on the Internet rather than negotiating. Enterprise software sales are rarely done that

way, but posting your price is honest and makes business sense (p. 895), so we do it anyway. That said, many founders act like tradition is there to be contradicted. Much of tradition is hard-won wisdom. So, traditional answers carry neither positive nor negative baggage; they should be investigated like all other ideas.

Constructive vs critical
It's easy to cut down ideas;[772] it's hard to create and execute them. Give me any idea and I can find someone who thinks it's dumb. So what? "Constructive" criticism means constructing, not just blasting. Look for advice with a clear method for implementation and a clear way to know whether it is working, especially since an idea worked for one company might not work for another.

Put your metrics where your mouth is
Does the advisor volunteer a clear way to measure (p. 645) the success of her new idea? If so, the idea is self-evaluating, and you can change if it happens to not work for you. This is the guiding principle behind our marketing efforts at Smart Bear.

Develop relationships
Actively develop a network of trusted advisors. These could be local entrepreneurs, on-line forums, even bloggers you like. Everyone needs a Gerry or two. Advisors won't always agree with each other of course, but nothing beats running ideas past people you respect and who truly have your interests at heart, even if their advice ends up being wrong.

Gut feel
Your gut often knows the answer. I know you don't want the answer to be "feelings," but sometimes feelings are wiser than thoughts (p. 159). In a world where both "X" and "Not-X" are convincingly peddled as The One True Way, you might need something outside of pure logic to resolve the path. If you find yourself vigorously agreeing with some new idea, that might be all the evidence you need.

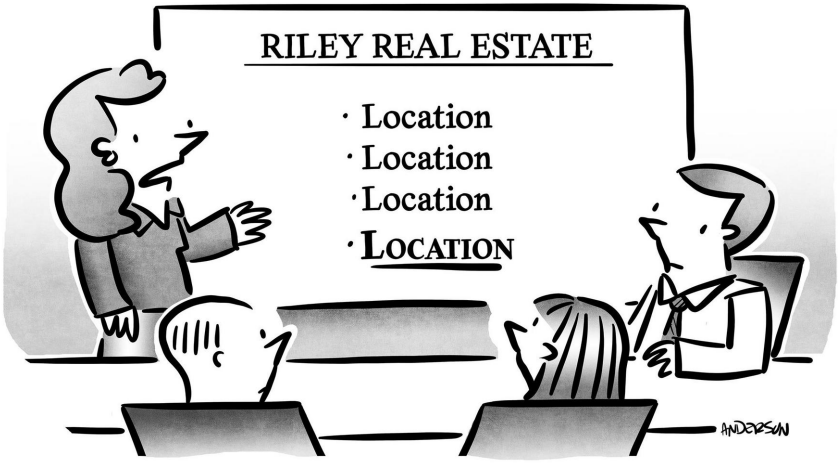
Besides, no one knows what's right, not even me (p. 449).

Seek advice that helps you become a better version of yourself, not advice that aims to change who you are.

Being yourself is the only thing you're going to be good at (p. 399), anyway, which is what will give you the leverage you need to win (p. 543).

Chapter 52:
Moats: Durable competitive advantage

TAXONOMY · EXAMPLE: CLOUD COMPUTING STRATEGY



"*This* is what separates us from the competition."

credit[773]

Economics is useless for predicting (p. 193) what the market will do tomorrow, but it is prescient at predicting long-term forces that shape entire industries. Business strategists ignore this at their peril.

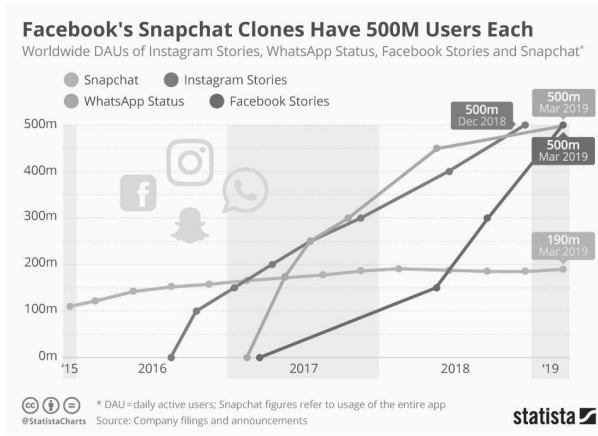
One of these forces is that industries commoditize: Companies copy the best ideas from each other, whether in features to attract customers or cost-savings to build profits, so they converge to similar products with similar cost structures. Undifferentiated in most dimensions, the flexible dimension is price, and in a real-life Prisoner's Dilemma,[774] price reduces until profits are driven to zero. It's great for the consumer—the best products at the lowest prices—but bad for the companies.

To avoid homogenization, a company needs differentiation that others cannot copy. Best is product differentiation that customers value, because that should lead not only to higher prices (because customers cannot get the same value elsewhere) but also greater market share. Second-best is differentiation in cost structure, because this allows the company to convert those savings either into profit (keep in the bank, or distribute by dividend or stock buy-back) or growth (spend in sales, marketing, or new product development).

Durable differentiation is rare, especially in the software industry where almost anything can be copied.

It's not enough to have a "competitive advantage;" the advantage must also be *durable*. Snapchat invented the idea of "Stories," but Facebook and Instagram copied it, so the advantage didn't convert into market share (Figure 1).

If "competitive advantage" isn't enough for durability, what is?



credit[775]

Figure 1: Facebook doesn't invent anything; they copy. But there is no "first mover advantage" unless that advantage is also durable, i.e. not easily copy-able by rivals.

A TAXONOMY OF MOATS

> *"The most important thing to me is figuring out how big a moat there is around the business. What I love, of course, is a big castle and a big moat with piranhas and crocodiles."*
>
> *–Warren Buffett*

Durable advantages are often called "moats," after Buffet. The concept appears everywhere—a characteristic of fundamental truth–

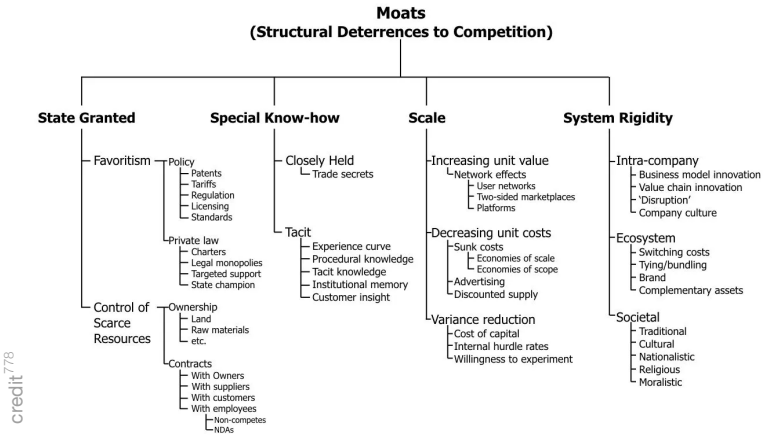


Figure 2

for example more recently and famously in Hamilton Helmer's Seven Powers.[776]

Jerry Neuman collected, organized, and detailed a terrific taxonomy of moats[777] (Figure 2). Such a reference is useful when analyzing other companies (which moats are they building?) or figuring out which moats are best for your business to construct.

A real-world example will breath life into these trite bullet points.

APPLYING MOATS TO A COMMODITIZED MARKET: CLOUD COMPUTING

AWS (Amazon Web Services) was the first and is still the largest "cloud computing" provider. Its industry is an archetypical example of a market that commoditizes. Indeed, all computing infrastructure

markets that preceded the cloud also commoditized, from individual components (e.g. RAM and disks) to entire physical data centers (with power, internet connectivity, climate-control, and physical security).

Cloud computing—paying only for what you need, by the hour, by the megabyte, scaling up and down at a flick of a switch—is a discrete leap forward in customer value, but commoditization is inescapable, as every by-the-hour-CPU, gigabyte-of-disk, and gigabyte-of-network-transfer from one vendor is the same as from another. If renting infrastructure is the only product, the only remaining dimension to compete on is price, so price falls, and the industry commoditizes.

Indeed, that's exactly what has happened, for example in storage (Figure 3).

But AWS has largely avoided commoditization. The proof is in the market-share and in the profit. Every year since 2014, AWS's profit has been more than half of Amazon's entire profit; in 2022 it's generating[780] $21B of profit at a 30% operating profit margin, and growing (Figure 4).

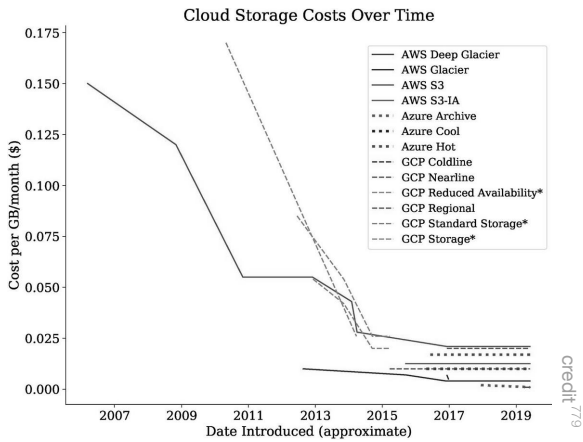


Figure 3: Different types of storage have different prices, but each type has identical price across vendors. All prices are low.

Figure 4

If it were possible for a competitor to steal business by undercutting prices, they would. Indeed, they *have*; competitors like Google, Microsoft, and Alibaba charge less for similar service. Google in particular spends billions more dollars per year than they make, plus they have all the requisite "smart people," but even at scale in 2022, on annual revenues of $20B for their cloud computing services, they're still *losing* nearly $4B annually. None of this is hurting AWS's profits, demonstrating that AWS has not commoditized (Figure 5).

Why not? We might expect the answer is "moats," but what are AWS's moats?

AWS Moats

In 2019, then-CEO of AWS (now-CEO of Amazon) Andy Jassy, on stage in a live interview with the great Kara Swisher,[783] was asked directly: Why hasn't AWS commoditized? Jassy immediately fired off the moats from their anti-commoditization strategy, annotated here with the bullet points from the taxonomy diagram:

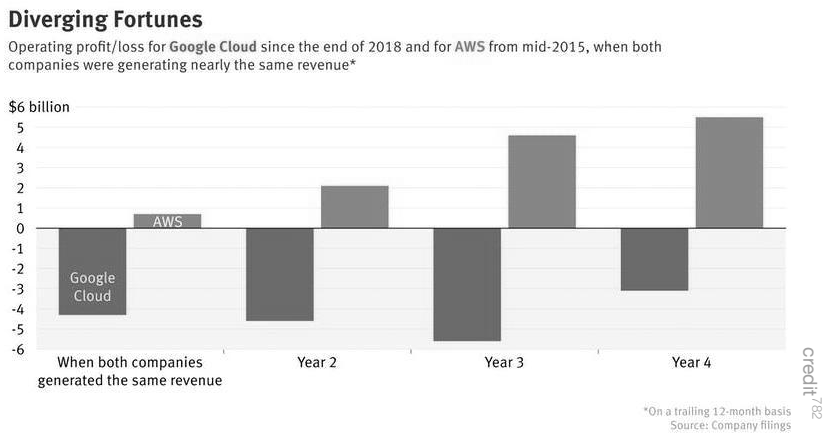


Figure 5

Innovates the fastest [Economies of Scale + Willingness to Experiment]
Jassy says they use their size to fuel more and faster innovation. Specifically, revenue today plus line-of-sight to revenue growth for tomorrow, means it is easy to justify a massive investment in innovation. The platform with the most innovation, will have the best tools for developers, which they believe is the key to winning.

Biggest Ecosystem [Complementary Assets]
The biggest vendor will attract the most 3rd-party support, which in turn is vital for developers who must integrate with other tools, especially in the modern world where everything is a SaaS.

Advanced operating maturity [Learning Curve]
Jassy says, "There is no compression algorithm for experience." Even Microsoft and Google, throwing billions of dollars and thousands of engineers at the challenges, are not able to overtake AWS.

There's another moat that Jassy didn't list, probably because he didn't want to talk about market-share,* and possibly because it could be interpreted as gently insulting to their customers:

> **Brand of "The Leader" [Brand]**
> The old adage was "no one was ever fired for picking IBM." Now that's true of AWS. The idea is that if you pick the market leader, your career isn't in jeopardy if it doesn't go well, because you picked the least-risky choice. Often "the leader" is also "more expensive," so this is not only good for long-term differentiation, but for profitability.

There's another moat that is clearly part of every major cloud provider's strategy; perhaps Jassy didn't mention it because it's slightly nefarious, as it involves lock-in (p. 275):

> **Has the biggest menu of technology to select from [Switching Costs]**
> If customers only buy commodity services like CPU, RAM, disk, and network, it's easy for them to leave for a cheaper cloud. So, AWS has created hundreds of "services" covering every imaginable corner in the world of infrastructure: big data storage and retrieval, web analytics engines, AI algorithms, logging, alerting, change-management, source code repositories, network firewalls, and support for dozens of open source software packages, all for rent by the hour or gigabyte. Furthermore, they encourage use of these services by making each one free when each is only lightly used. A software developer who avails herself of this incredible array of services is more productive than a large team of developers would have been ten years ago, however her Faustian bargain is that she gives up her ability to switch to a competitor, which means as her project grows and the pricing tiers kick in, she's locked in.

---

* Later in the interview, he refused to answer basic questions about how much market share AWS has, or the relative positions of Microsoft, Google, and Oracle, even under pressure from Kara Swisher, the excellent interviewer. Being the market-leader invites regulation and puts you on the defensive in anti-trust cases, so public companies shy away from such claims; for example Amazon owns 43% of all eCommerce at the time of this writing, but they publicly position themselves as a general "retail" company, so that their share is "only" 6%.

MOATS ARE INTEGRAL TO STRATEGY

Moats don't appear quickly or by accident; they require consistent investment and prioritization over a course of years. This is exactly what a "strategy" is supposed to organize, and what "just be agile and react to whatever happens" will not. Therefore, a good strategy will have identified one or very few moats that the company will create, and the primary themes of action that will excavate them and fill them with Buffett's piranhas.

The example of AWS shows that the taxonomy is only a starting point. The details of exactly how one item applies to a specific company are essential. A strategy that states "We will create switching costs" is not a strategy; it is a statement of what you hope will become true, rather than an explanation of how you will cause that to become true. A strategy to intentionally create switching costs could look more like this:

Objective
Create permanent customers through high switching-costs, caused by extensive integration with their technology stacks, development workflows, and IT administration.

How
Maximize the number of cloud services each customer uses. This, in turn, requires:

1. Create the most number of services as quickly as possible, so there's more to integrate with. Must have more than the competition, and a reputation of "innovating faster than anyone else," to attract customers in the first place, and so that customers don't look elsewhere if we don't have their favorite service quite yet.
2. Create compelling "on-ramps" so that services are actually used (e.g. free tiers, great documentation, pre-integration with other APIs, "bundled offerings" that includes many APIs)

Consequences

1. If we build many services, quickly, some of those services will turn out to be duds, or might be strategic only as a part of the larger offering, not high-growth or even profitable on their own. This is a "cost of doing business" and will not constitute evidence that a team has failed (p. 1261) or that the strategy is failing.
2. Many individual services will be commoditized, especially when we're just reselling existing open-source projects. We still need to create these, because integrating many of them into a single customer project is necessary to fulfill the objective, and customer discovery shows that they specifically want to buy those products from us. Creating genuinely unique services are more valuable and defensible, so we need to do that as well, but they also take 100x more effort.
3. Given that (1) and (2) demand a large quantity of software, built quickly, this is a costly bet; we require $X billion dollars over three years, plus a scaled-up hiring process.

The moat is named, but also how it will be built, along with second-order consequences, even *undesirable* consequences (high-cost, and

frequent failures of usage or of differentiation on a case-by-case basis) that we not only *accept*, but *expect*, in service of the objective of creating that moat. With these details and trade-off decisions in place, it is a strategy, not a hope.

Make sure your strategy is explicit about which moats are being constructed, and the major multi-year activities required to construct them.

credit 784

"I told you to order the piranhas earlier."

Chapter 53:
The fundamental forces of scale

TWO FUNDAMENTAL CHALLENGES · ROBUST TEAMS
ROBUST SYSTEMS · PREDICTABILITY · MATERIALITY
RECRUITING · COMMUNICATION
TECHNOLOGY & INFRASTRUCTURE
RISK-MITIGATION · LARGE ≠ SMALL × 100

"I'm not letting myself go, I'm scaling."

credit785

Idealistic founders believe they will break the mold when they start scaling. They will *not* turn into a "typical big company."

By which they mean: No stupid rules that assume employees are dumb or evil, with everything taking ten times longer than it should, with wall-to-wall meetings, and hiring "average" or "normal" people.

That is, preserving the positive characteristics of tiny organizations while avoiding the common problems of large ones, by adapting the startup's existing processes, that have served them so well thus far. We're merely doing it with more people now, and still figuring it out as we go along, exactly as we always have.

Why do they never succeed? Why is it impossible for all of these intelligent, well-meaning founders to run a 500-person organization like a 50-person one? What are these unavoidable forces?

Are they really unavoidable?

THE TWO FUNDAMENTAL CHALLENGES OF SCALE

After understanding these two primary drivers, we'll see how they explain the challenges that arise in every corner of the business.

Rare things become common

At scale, rare things become common (p. 1345).

Rare things are difficult to predict, and typically difficult to prevent. Before scale, they are, well, *rare*, so this doesn't matter; they can be handled manually. But when they start happening daily, new processes become necessary.

For instance, with 1,000 servers, a server failure every three years translates to one failure per day. This frequency impacts not just DevOps but also customer support and social media teams. Customers are impacted and complaining every day, even if each individual event is rare, impossible to predict, and impossible to prevent. Automation helps, but doesn't solve the problem.

This is inescapable math crops up all over the business.

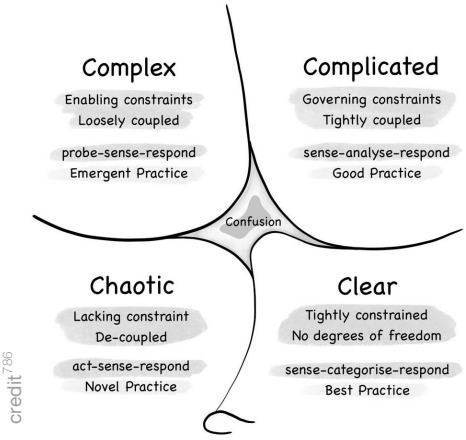
Complexity beyond human comprehension

Human minds have limited computational abilities. A sufficiently complex system cannot be fully comprehended at all.

"Complicated" systems can be comprehended. These types of systems are intricate, but they exhibit properties which allow people to tame the beast. Components of complicated systems can be separated, built and tested on their own, then assembled; thus, divide-and-conquer is a useful technique. Challenges are difficult but are solvable, especially if you have seen them before; thus, "experts" and "specialists" can solve the puzzles. Repeatable processes can monitor the system.

"Complex" systems, in contrast, have components which affect each other, often cyclically. Therefore, while it's still useful to build and test components separately, there's just as much complexity in their interaction. Indeed, most of the difficulty lies in those interstitial spaces. The human brain is composed of neuron "components" which are relatively well-understood; it's in the complex, multi-directional, cyclical interactions that everything interesting—and inscrutable—unfolds. Large-scale effects like "consciousness" completely evade our understanding, as it emerges from complexity we have yet to comprehend.

As a company scales, there are both kinds of systems, but the complex ones are fundamentally difficult, and never stop being difficult. Their difficulty grows non-linearly as the company grows.



This "Complicated" vs "Complex" terminology comes from the Cynefin Framework.[787]

And even our unthinkably complex human brains, cannot comprehend the system.

With the two fundamental principles in hand, we can detail specific challenges of scale.

THE HUMAN COST OF ROBUST TEAMS

A "team of one" is the fastest, most efficient team, if you measure by "output per person." There's no communication, no meetings, and de-

cisions can be made instantly. Small companies operate this way by necessity, and it works! It's a big reason why they move quickly, and "punch above their weight" as is often said.

But, an illness takes the velocity of the product or the quality of customer service from heroic to zero. With a small team, if one person leaves, you've just lost six months to hiring and getting-up-to-speed on that project. Or twelve months or a complete rewrite because there wasn't any processes and documentation in place... because it was just one person, who didn't need that stuff, because after all we're moving so quickly! And not communicating!

Or it's fatal because that was a co-founder. "Founder trouble" is a leading cause[788] of startup death.[789]*

So **a team of one is fast, but brittle**. When you're small, this is a good trade-off, because speed is critical for combating the things that are constantly about to kill the company—lack of customers, lack of market attention, lack of core features, all the things (p. 71). When you're large, with 15-25% annual employee turnover, not to mention illness, vacation, and family, people will be leaving (temporarily or permanently) every day—rare things becoming common. Maintaining the same attitude as when you were tiny is disorganized and irresponsible, and the company would fail to function.

So, at scale, no project can have fewer than, say, three people dedicated to it, plus management and possibly some form of Product or Project Management. But that team of 4-5 will not be 4x-5x more productive than the one-person team; **per-person productivity goes down in exchange for robustness and continuity**.

On the other hand, while the small company loses 9 months to the loss of a key employee, or even implodes, the big company is the steady turtle that adds thousands of customer per month like clockwork and wins the race.

---

* Although data also show that companies with only one founder are more likely to fail.[790] So which is better? To me, it comes down to temperament—some people really want to be the only one at the top, and some want to share the burdens with others.

credit.791

"I dunno, 'first mate' has been done to death. How about this -- '*Co*-pirate?'"

THE HIGH COST OF ROBUST SYSTEMS

You might think that software systems wouldn't suffer the same brittleness as human systems, because we have things like automation and DevOps. Sadly, robust software is similarly much more costly and complicated.

Consider the case of serving a website (don't worry, you don't need to be an software engineer to follow this example!). Suppose there's a network connection to the Internet, a server that runs the web site's software, and a database that holds the web site's content. We have a single server, and it works most of the time. Let's say it works 99.9% of the time; that's sounds pretty good!

Well, it's not that good, actually. Failing for 0.1% of a year means it has failed for more than 500 minutes—surely an unacceptable amount

of down-time! This happens because a rare thing (0.1% failure) became common (at the scale of about half a million minutes in a year).

What if we had two servers instead? That way, when one fails, the other is still available. They'd *both* have to fail *simultaneously* for the website to be offline, which would happen `0.1% × 0.1% = 0.0001%` of the time, which would be only 30 seconds per year, so that really *is* great.*

OK, that's twice as expensive, because we now have two servers, but it's worth it for being more robust.** But wait, that's not enough, because the traffic that comes in from the Internet needs to be distributed across the two servers, and it needs to have some intelligence to route traffic to the healthy one, in case one is failing. That's called a "Load Balancer," and all cloud infrastructure providers have them, and they cost more money. So that's another component to manage, and more expense.

And wait, there's also that database. The two servers need to share a common database of content, so the database needs to be moved off to its own server. And what if *that* server fails? Then we have down-time again. So we also need two of *those* for continuity.

Now we're up to four servers plus a Load Balancer, and more than 4x the original cost. You can buy robustness, but it's far more expensive than you might think. We haven't even gotten to the fact that managing multiple servers is more difficult, or that there can be failures in the communications between servers, or failures in the Load Balancer which takes down the whole system***–all decreasing our robustness, requiring even more effort to deliver the end result.

---

* Sadly this is not really how the world works; typically failures are correlated or cascade. We'll keep things simple for sake of the example, but this complication proves the point yet again.

** You might think we can save money by buying cheaper servers, because each needs only one half the capacity of the original. But no, because in the failure mode, a single server still needs to serve all the traffic, so each needs to be powerful enough to work on its own.

*** Yes, there's solutions to this too—more redundancy at the network layer, often with a different vendor, but again, all more cost and complication!

Robustness always carries a cost. Whether it's running four servers instead of one, or a team of four instead of one, you trade local inefficiency for whole-company continuity.

PREDICTABILITY (THAT YOU ACTUALLY NEED)

When you're small there's no need to predict when a feature will ship. Marketing isn't scheduling a launch and Recruiting isn't timing the start-dates of the next 50 hires in customer service and sales. This means you can—and should!—optimize myopically for speed-to-market.

Small companies correctly brag about their speed as an advantage, but it's easy to see how a larger company has a different—and massive—advantage of execution. Sure, when our company WP Engine launches a new product, the marketing department needs predictability for the launch date, but that's because it's a highly-skilled, well-funded, coordinate group of teams that explode with press, events, campaigns, social media, and newsletters, grabbing more attention in a single week than a smaller company might scrape together in a year. There are also global Sales and Support teams, so we're immediately selling to 200,000 existing customers as well as thousands of new customers per month, which means we'll add more additional revenue each month than a small company will earn over a whole year.

But all this requires predictability. We didn't line up that press and have those sales materials and train hundreds of support reps and ensure that code-quality is high enough to scale on day one, without predictability. Predictability requires going slower. Predictability requires estimation (takes time), coordination (takes time), planning (takes time), documentation (takes time), and adjusting the plan across

"AnderCorp surprised investors expecting the worst today by performing completely miserably."

credit[792]

all teams when life inevitably unfolds differently from the prediction (p. 193) (takes time).

When you're hiring lots of people, you need predictability. Consider the timeline of adding a technical support team member. First, Recruiting is casting about for potential candidates. Then scheduling and performing interviews. Then waiting for them to quit their job and take a week off. Then new-employee-orientation. Then classroom training. Then paired up with senior folks on the floor as they ramp up their skills and comfort. Therefore the time between deciding to grow the support team, and having new people up-to-speed, is four to six months.

Therefore, we have to predict the demand for technical support four to six months in advance, because we need to be hiring for that *right now*. If we under-estimate, our support folks get overwhelmed with too much work; their quality of life suffers, and service to each customer suffers. If we over-estimate, we have too many people, which

is a cost penalty.[793] Of course the latter is a better failure mode than the former, but both are sub-optimal, and the solution is predictability.

"The future is inherently unpredictable (p. 193)," insists the small company, spurred on by Lean and Agile mindsets, and the truth. Indeed, blue-sky invention and execution are unpredictable. But this is also a self-fulfilling prophecy; to insist the future is unpredictable is to ignore the work that could make it more predictable, which makes it in fact unpredictable *to that person.*

Small companies don't have the data, customers, institutional knowledge, expertise, and often the personal experience and skill set to predict the future, so they are usually correct in saying it's impossible. But is it impossible in principle, or is it impossible for *them*? At scale, it becomes required. Not because Wall Street demands it, nor because investors demand it, nor any other causal derogatory excuse made by unpredictable organizations, but because it's critical for healthy scaling.

credit[794]

"But, to be fair, there's a fifty-percent chance of just about anything."

THE MATERIALITY THRESHOLD

If Google launches a new product that generates $10,000,000/year in revenue, is that good? No, it's a failure (p. 1261). They could have taken the tens of millions of dollars that the product cost to develop, used that to make their existing operation just 0.01% more effective, and made far more money.

At more than $100B/year in revenue,* Google can only consider products which have the potential to generate $1B/year in revenue as an absolute floor, with the potential to grow to $10B/year if things go better than expected. Things like YouTube, Cloud, and self-driving cars.

This principle is called the "Materiality Threshold," i.e. the minimum contribution a project must deliver, for the project to materially affect the business.

With a small business, the materiality threshold is near $0. A new feature that helps you land just a few new customers this month is worth doing. A marketing campaign that adds two sign-ups/week is a success. Almost anything you do, counts. That's nice—it feels good to be moving forward.

The financial success of the larger company dictates a non-trivial materiality threshold. This is difficult. Even a modest-sized company will need millions in revenue from new products, maybe tens of millions in the optimistic case. Very few products can generate that sort of revenue, whether invented by nimble, innovative startups or stately mature companies. As proof, consider that the vast majority of startups never reach a $10M/year run-rate, even with decent products and extraordinarily dedicated and capable teams.

---

* *Editor's note:* This was originally written in 2017; in 2023 Google's revenue is nearly $300B, which reinforces the original point; they could not have achieved that growth if they worked on projects that generated revenues of "merely" $100M.

Yet, it's the job of a Product Manager at that mid-sized company to invent, discover, design, implement, and nurture exactly those products—something that most entrepreneurs will never succeed at. Tough job (p. 817)!

RECRUITING

Employee #2 will join a startup for the experience (p. 1417). Even with a significant salary cut, and even if the company fails—the most likely outcome—it's worth it for the stories, the influence, the potential, the thrill, the control, the camaraderie, the cocktail-party-talk.

Employee #200 won't join for those reasons. Employee #200 will have a different risk-profile regarding their life and career. Employee #200 will be interested in different sorts of problems to solve, like the ones listed in this article instead of the ones where you're trying to understand why 7 people bought the software but the next 3 didn't. Employee #200 will not work for a pay-cut.

Small companies view this as an advantage, and certainly it's advantageous to recruit amazing people at sub-market rates. But there hundreds of employees at WP Engine today who are much more skilled in their area of expertise than I've ever met at a small startup, including my own. Why?

One reason is that people with a lot of experience are often in a different phase of life, where family and other demands means they want a predictable, larger paycheck for a predictable, well-defined job. When you combine their advanced skills with the raw materials of a scaled company (i.e. customers, brand, funding, teams that can execute larger ideas), they can generate huge value while still tucking their kids into bed at night.

credit[795]

"So, where do you see yourself in ten minutes?"

Another reason is that, after developing that expertise, they find it's enjoyable to apply their skills within a larger environment. For example, there are advanced marketing techniques that would never make sense with a smaller company, that are fascinating, challenging, and impactful to the top line at a larger company. There are talented people who love that challenge and would hate going "back to Kindergarten" as Jeff Bezos famously quipped, scratching out an AdWords campaign with a $2000/mo budget (p. 1369) or assembling the rudiments of SEO or just trying to get a single marketing channel to work[796] or being called a "growth hacker" because they finagled a one-time bump in traffic.

This has implications on compensation, how you find that talent, and why that person wants to work at your company instead of the one down the block who can pay a little bit more. Therefore, it's critical to have a mission that is genuinely important, have meaningful (p. 399) and interesting work to do, connect everyone's work to something bigger than any of us (p. 827). These matter even more at scale, because it's the reason why talent will join and stay.

credit[797]

"I don't get it! We've got a mission statement, a credo, *and* a mantra!"

COMMUNICATION

With four people in a company, any information that everyone needs to know can be told to just three other people. Everyone *can* know everything. If there's a 5% chance of significant misunderstanding, that doesn't happen often.

With four *hundred* people, it's never true that a piece of information can be reliably communicated, in a short period of time. A 5% chance of misunderstanding means twenty people are confused. That's assuming they even read the communication, or listened during the entire presentation over Zoom. What's the chance of *that*? Don't ask (p. 997).

"Slack" is not the answer.[798] "Email" is not the answer. The answer is: **Repeating simple messages.**

"Repetition" is the answer to "I didn't see it." Repetition in different formats, at different times, by many leaders. Asymptotically achieving 100%, although also creating collateral damage for the people who really do read and listen to everything, who are tired of hearing the

same thing, and wondering why they're being punished for actually paying attention.

"Simple" is the answer to "I didn't understand / remember it." Just as with simplicity in strategy (p. 489), you have to accept that people don't read, people don't remember, people have other things on their minds, people don't understand language as well as you wish, and you didn't write as clearly as you hoped.

But doesn't this mean you cannot communicate anything complex to 1000 people? Yes, that's what it means. Scale is hard.

TECHNOLOGY & INFRASTRUCTURE

Managing 10,000 virtual servers in the Cloud Era sounds easy. Automate everything, then any process that works for 100 servers, will work for 10,000 servers just by doing the same thing repeatedly—exactly the thing computers are excellent at.

It never works like that. Reddit took 18 months[799] to get "number of likes" to work at scale. StackOverflow took 4 years[800] to get everything converted to HTTPS. Wired did that conversion[801] in a "mere" 18 months. Everything is hard at scale.

What are the patterns in those stories?

One is another application of "rare things are common." Rare things are hard to predict and can be hard to prevent. Often they're hard to even identify and sometimes impossible to reproduce. This is fundamentally difficult.

Another is continuity or compatibility with existing technology. New companies get to start from scratch, but at-scale companies must transform. New companies like to make fun of large companies for how hard it is to transform, neglecting that the cause of the difficulty might also be generating $100,000,000 in revenue this year.*

Another is bottlenecking. All hardware and software systems have bottlenecks. At small scale, you don't run into any bottlenecks, or at least the ones you do can be solved with simple techniques like increasing capacity (p. 945). Eventually something difficult breaks and you have to rearchitect the stack to solve it. Even something simple like converting HTTP links to HTTPS or updating "number of likes" in real-time, becomes a monumental architectural challenge.

Another is Hickam's Dictum:[802] Problems in complex systems generally have more than one root cause. Consider the case of a software upgrade causing problems for seven customers, in a way that wasn't detected. What is the "root cause" of this problem?

- The problem is we didn't detect the failure; had we done so, we could have reversed the upgrade immediately.
- The problem is we didn't test the thing that failed; had we had more tests, the problem would never have reached customers.
- The problem is we didn't document the code that broke; had the code been clearer, the human wouldn't have coded the error.
- The problem is we didn't review the code properly; had the code been reviewed with a proper checklist, we would have discovered the bad code before it was deployed.

As with Five Whys,[803] there are always deeper reasons, as well as different reasons. Different from Five Whys, there isn't just one that is the "root" cause; instead there are a lot of things that interact with a lot of things. This doesn't mean it's hopeless, it just means it's complex to analyze, complex to decide what to do, and that Occam's Razor doesn't apply.

All this slows down development and adds investment. There will be entire teams who focus on infrastructure, scaling, deploys, cost-management, development processes, and so forth, none of which are

* Instead of "legacy code" we call it the "revenue code."

directly visible to or driven by the customer, but which are necessary to manage the complexities of scale.

RISK-MITIGATION

For a small company, the most likely cause of death is suicide.[804] Usually it's starvation—can't get enough customers (distribution) to pay enough money for long enough (product), in the cascade of things-that-must-be-true for product/market fit (p. 71). But also things like founders splitting up, not getting enough traction to self-fund or to secure the next round of financing, having to go back to a day job, and so on.

At scale, the risks are different. There is very low risk that WP Engine will not sign up thousands of new customers this month. Other risks, however, are not only possible, but likely. Addressing those risks head-on, is required for a healthy and sustainable business that can last for many years.

Take the risk of business continuity during a disaster scenario. What if an entire Amazon data center became disabled for a week? How quickly could we get all those customers back up and running? Would that be true even though thousands of other businesses are also trying to spin up servers in other Amazon data centers at the same time? Could we communicate all this with our customers quickly and simply, so that our support team isn't overwhelmed by repeating the same message to tens of thousands of justifiably-angry customers?

Risk-mitigation can even result in growth. Serious customers want to work with vendors who understand and mitigate risk; this maturity becomes a selling point. That's why enterprise suppliers (like WP Engine) are constantly flouting their compliance with SOC 2 and ISO 27001 and all the others. Small companies make fun of those things

credit 805

"It's important to remember that correlation does not imply causation. Besides, we all know it was Brian."

as being unnecessary at best or a false sense of security at worst, but while they're busy making that point, the larger companies are busy signing three-year multi-million dollar deals.

Early on, you do not need a disaster-recovery plan. That won't be the thing that will kill the business, and your customers will understand if a young business is subject to that sort of risk. Later on, this becomes critical, and worth investing in.

LARGE COMPANIES ARE NOT SMALL COMPANIES × 100

These forces cause larger companies to be fundamentally different than small ones. It isn't a bad thing or a good thing. It's a different thing.

Some idealistic founders believe the root cause of scaling issues is the "command-and-control" organizational structure. But none of the

examples above make reference to any organizational structure. It's universal. This is why Holacracy[806] and Teal Organizations[807] do not solve these problems. It could be a fantastic idea to experiment with organizational structure, but the fundamental forces above will not be eliminated with a recombination of roles and power structures.

Scaling is hard, the road is foggy and bendy, it lasts for years, the set of people you need might be different, and no one emerges unscathed. So, it is not a sign of disaster if you have difficulty wrestling with these forces. Everyone does.

Disaster is when a company is scaling, but the leaders don't appreciate these forces, don't work constantly to morph the organization accordingly, don't bring in experienced talent, decide they can figure it all out as they go along without help. Rather, it should mean new people, new roles, new values, new processes, new recruiting, new stories, new constraints, new opportunities.

Too many founders and leaders want to believe:

> What got us here is what's important and unique about us, and thus we should preserve *all* of it. Other companies fail because they "act like big companies," but we'll avoid all that because we're smarter than they were. As evidence of our acuity, just look at our success thus far. We will continue to succeed in the future as we have in the past.

But they're wrong.

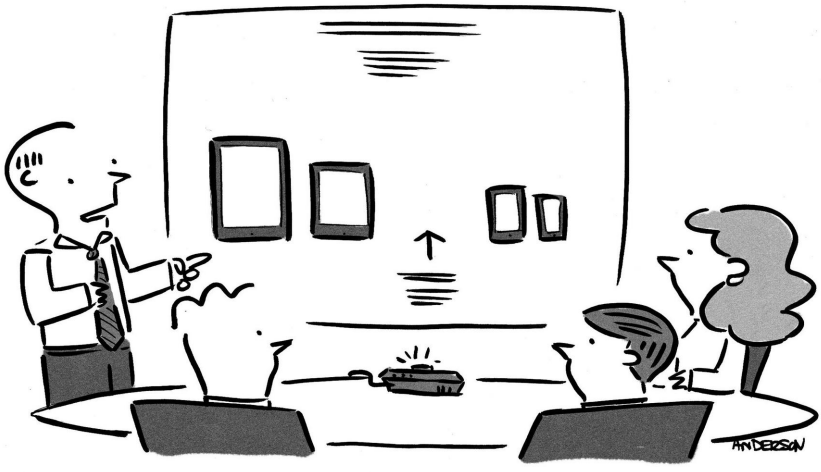
There should be a few values that are kept constant, that's true. Otherwise none of it means anything (p. 827). But the details must change (p. 1299).

Many founders and leaders can't make the shift. This always hurts the company, and sometimes kills the company. The world is full of those horror stories. It's sad, because it's an avoidable waste of opportunity and sometimes hundreds of person-years of effort.

Don't become one of those cautionary tales.

Chapter 54:
Adjacency Matrix: How to expand after PMF

EXPANSION · ADJACENCY
THE ADJACENCY MATRIX
SELECTING THE BEST OPTION



"We believe there's room in the marketplace for a revolutionary new device somewhere between the 7-inch mini tablet and the 6.3-inch mega smartphone."

EXPANSION

Our everyday craft of incremental product evolution is obvious and natural.

We hear from customers in emotionally overcharged outbursts on Twitter, or public reviews encoding their ire or gratitude, or filtered through the problem-solving of tech support, or teasing us with an if-you-build-it-we-will-come sale that hasn't yet (and may never) close. We then convert what we thought we heard into features and bug-fixes, prioritized by impact and growth ideally, but more likely by pride and intuition; sometimes we get lucky and those are the same thing.

This works, after Product/Market Fit (p. 335). That is, once everything is fundamentally working, and the job is "don't break what's working." Then we should incrementally add utility and delight (p. 275) while not disrupting the money-making flywheel. Before PMF, it's not working yet, so the job (p. 9) is to identify what can work (p. 71), and for whom (p. 317), not to make slight adjustments to something that's not working.

But then we come to another post-PMF decision: **When and how to expand the scope of the business.**

Not a *pivot*—that's for companies that are staring death in the face, that must strike out in a new direction for a chance at survival. Rather, an *expansion*—to keep what's working, but sprout new shoots in new directions. Adhering to the rules of great strategy (p. 489) by leveraging existing assets (p. 543) to attack a valuable new opportunity with asymmetric upside. (Add your favorite buzzwords; those were mine.)

When is the right moment to venture into new territory, rather than building obvious, incremental things with little risk?

Bottlenecked growth
If existing marketing and sales channels are at their limit, i.e. when spending more time or money doesn't yield more output, or at least, not cost-efficiently. Then an expansion to a new channel, or new geography, or new market segment, accelerates growth.

Target market saturation
If you've already won 5% or more of your perfect market. Some people use acronyms* to name this concept; I mean of the total number of people who are your ICP (Ideal Customer Profile) (p. 317)—oh goodie, another acronym—the absolutely-perfect-in-every-way customer whom you are targeting with all your marketing and product features. 5% is not saturation, but if you got to 5%, you can presumably get to 10% with similar methods (p. 839). Since you're well on your way to that already, you can afford to turn your attention to segments that are similar enough to be relevant, but different enough that they require effort to serve well.

Diversification
Adding new marketing channels, adding new target markets, adding new geographies, adding new pricing options, should not only add growth, but make the company more robust to market and economic disruptions. At WP Engine we experienced this after expanding upmarket. It wasn't just "growth"—addressing a new segment meant our value proposition became different. Small businesses saw us as "**expensive**, but the best," whereas large companies saw us as "**low cost**, yet still enterprise-grade." This diversification of positioning resulted in robust growth during the COVID crisis, where smaller companies were going out of business (i.e. "expensive" is now bad), but larger companies were looking for ways to save money (i.e. "low-cost" is now good). When one segment has difficulty, the other grows; the net effect is stability. At scale, this is nirvana.

When you can afford to invest
A company throwing off more than a million dollars a year in profit can afford to try something new; failure in a new investment is not fatal. Expansions are both costly and risky; extra money covers both liabil-

* TAM (Total Addressable Market), or SAM (Serviceable Addressable Market); there's more.

ities. Extra money could come in the form of outside investment or profit draw-downs, but either way it's a bet, it's an investment, and the typical investment rules (p. 867) apply.

If expansion is warranted, we come to the crux of this article: **How do you decide in which direction to expand?**

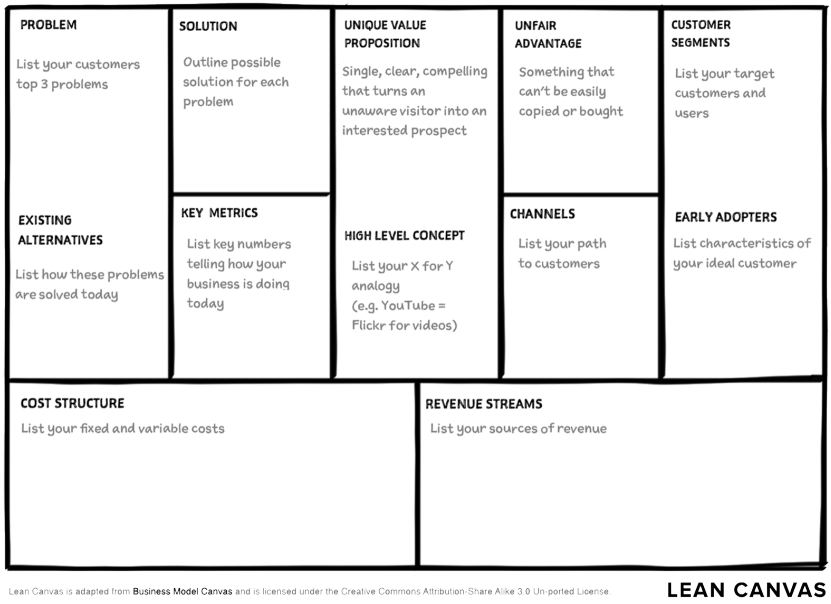
ADJACENCY

The key idea is "adjacency," meaning "close by." The difference between incremental change and expansion is that expansion is "somewhere else," but the difference between expansion and something too far afield is "adjacency."

You don't want to go so far afield that you're taking on too much risk, you're not leveraging existing assets, and thus it is too risky, and too costly, such that even if the opportunity is large and tangible, it's still a bad idea *for you*. A great strategy, that doesn't align with your strengths, is a bad strategy *for you*.

It's not hard to think of ways to expand the business. You could enter a new geography. You could target a different ICP, different niche, different vertical, different use-cases. You could go up- or down-market in the size of business you sell to or in the customer's Needs Stack (p. 259). You could add a freemium tier, add a higher pricing tier, add a lifetime plan, add a new product line. You could expand from iOS-only to add Android, from web-only to mobile-app, from integrating with one third-party to ten. You could venture away from the reputational investment of SEO and social channels into the analytical world of paid-marketing, or venture from the pay-for-today world of advertising to the invest-in-the-future world of SEO and social.

If you want to be more systematic, you can use something like the Lean Canvas to ideate (Figure 1).



credit[809]

Figure 1: Ash Maurya's[810] version of the popular business model "canvas."

Fill each box according to the existing business, then brainstorm* ways that you could expand or improve on each box.

Calculating the potential upside of these ideas will of course be specific to the idea and the company. In this article, I give my general framework for evaluating the cost and risks of executing those ideas, answering the question: **How adjacent is this idea** to the current business?

* Here's some fresh and fun ways to brainstorm (p. 53) these sorts of ideas. LLMs can be helpful—not to select ideas, but to generate possibilities.

THE ADJACENCY MATRIX

Start with a table of the major functional areas of your business. It will be similar to the following, but tune it for yourself. For example, a company leveraging product-led growth may have no Sales department, but might have a heavy design culture and thus adds a row for Design.

Functional area	**Types of activities & responsibilities**
Marketing	getting attention; brand personality & positioning; acquisition channels
Sales	processing leads; managing the sales process; pitch & competitive materials; domain expertise; sales training
Service	expertise; technical knowledge; domain expertise; product training
Product	understanding the customer; mix of data and intuition; roadmaps; everything else (p. 817)
Engineering	platforms; architecture; major libraries & frameworks; tech debt; infrastructure; specialized skills and knowledge
Business Model	Pricing and packaging; unit economics; profitability; budgeting

Now evaluate the "adjacency" of your proposal by determining **how much each of these activities must change** to support the new initiative. Your only choices are:

- ✅ **Trivial**—Almost no changes required. "Training" is an email or one Zoom session. Adjusting sales and marketing material means adding a few bullets.
- ⚠️ **Adjustment**—Change-management required, but manageable within current processes, norms, and organizational structure, similar to a "large new feature release." Requires training. New sales slides. A new web page for the website. A non-trivial change to pricing and packaging. New SMEs[811] in Support.
- ❌ **Overhaul**—Major change needed: hiring for new skills or for capacity, significant retraining that might require new specializations, structural process or management changes, especially if the org chart is changing. Often you're not entirely sure of the full extent of the changes, i.e. it is sufficiently complex that we can't identify all the challenges and risks that await us.

Longtime readers will recognize this as another instance of my Fermi Estimation* hobbyhorse, in which we intentionally limit choices to avoid arguing over details and predictions that we aren't qualified to make anyway (p. 193). Dispositioning into one of these three buckets should be relatively easy, without extensive analysis.

As an example, I'll take our company WP Engine in 2013 when our ICP (p. 317) was "small to mid-sized company home pages and small to mid-sized blogs/media, often built by freelancers and small agencies." We'll evaluate two ideas: (a) expand to support marketing campaigns run by Enterprise-sized companies, and (b) expand to host the main websites of Enterprise-sized companies:

* Described completely in how to use Fermi estimation for ROI-type decisions (p. 171), including both objective dimensions like "time" and subjective dimensions like "what makes for a compelling product," and echoed in pieces on probabilities (p. 997), evaluating markets (p. 71), and making big decisions (p. 603), among others.

Functional Area	Enterprise Campaigns	Enterprise Full Site
Marketing	⚠️ **Adjustment** (new target audience, and message of "campaign" instead of "home page")	❌ **Overhaul** (completely new competitors, new marketing channels, establish brand from scratch)
Sales	❌ **Overhaul** (drastically new sales processes and cycle times)	❌ **Overhaul**
Service	✅ **Trivial**	❌ **Overhaul** (e.g. new people and specializations like account management and white-glove on-boarding)
Product	✅ **Trivial** (campaigns don't require new features)	⚠️ **Adjustment** (new compliance requirements, but few new features)
Engineering	✅ **Trivial**	✅ **Trivial** (enterprise websites get similar amounts and types of traffic as popular websites belonging to small companies)
Business Model	✅ **Trivial** (existing plans are sufficient)	⚠️ **Adjustment** (new plans, but same business model)

In this case, the conclusion might seem obvious—of course it's easier to serve a small use-case in a new segment ("Enterprise") than it is to compete in a major use-case in a new segment.

What might not have been obvious is how dramatically different it is to "sell to the Enterprise." Often startups claim this as their growth path, even when they're at only $500k in ARR. This is definitely the wrong strategy at that moment; this exercise makes it clear, yet companies often conclude the opposite. They should instead be con-

sidering simple use-cases at larger companies, or they should be ignoring the complexity of large companies so they can continue winning where they are already strong.

Furthermore, it shows how much investment is needed if you insist on investing in Enterprise. Sometimes that *is* the right strategy. After all, calculating "how adjacent" is about *evaluating* the decision, not *dictating* it. If, for example, a company is at $60M ARR with steady growth in absolute dollars but slowing as a percentage of revenue, spending $10M to add an Enterprise-focused business model could be a great growth strategy.

In particular, when more than one area requires a "full overhaul," that's a deal-breaker if this is supposed to be an incremental, sustaining innovation. If multiple areas require a full overhaul, this is only acceptable if (a) you are willing to make a huge investment and (b) you're willing to take a large risk that it will not pay off, and this only makes sense if (c) the potential upside is enormous. This reiterates the first rule of investments (p. 867).

At some point, the idea is so non-adjacent that it's definitionally a bad strategy to attempt it, regardless of upside. Strategy is supposed to leverage existing assets; don't select something that doesn't do that.

SELECTING THE BEST OPTION

It is tempting to use a value-divided-by-cost analysis to decide which idea to select, where "value" is some estimate of upside, and "cost" is some formulaic summary of the Adjacency Matrix, or possibly an actual dollar or time estimate.

If you are so tempted, I recommend this ROI system (p. 171), which will force you to be rough-shod in both variables, ideally bringing the best ideas to the top of the list with a minimum of debate.

credit[812]

"Sometimes I look back and wonder what my mother, a simple dairy cow, would have made of all this."

However, whenever you are making an investment (p. 867)—doing something with substantial cost, time, risk, and only hopeful upside—I recommend solving first for maximal impact, and only secondarily for cost. The reasons are given in that article specifically in the context of investments, and also in my work-prioritization system that extends the Rocks, Pebbles, Sand analogy (p. 221). You might want to use Bin-stack (p. 603) to identify the items of highest value, only then looking to cost and risk to break ties.

In this context, the Adjacency Matrix is useful for completely ruling out ideas that are clearly too far afield, or identifying those which are particularly low-risk.

The Adjacency Matrix outlines your cost and risk analysis, identifying the areas of the business that must be considered. For changes that you already understand, you can describe what needs to be done, what the budget is, and what risks remain. For complex changes, especially the dreaded "unknown unknowns" likely to rise from Overhauls, you'll need to include more: the mitigations will you

put in place, objective measures or milestones to catch problems as early as possible, how you will attack the high-risk, high-uncertainty areas first, and specialized hires who have seen this movie before.

Finally, **the Adjacency Matrix is useful in communicating the decision** to the whole company—something leaders perennially fail to adequately appreciate and value. It's vital that the decision is simple to explain and justify, so everyone feels that it's natural, intelligent, and clear. The matrix can help form the narrative:

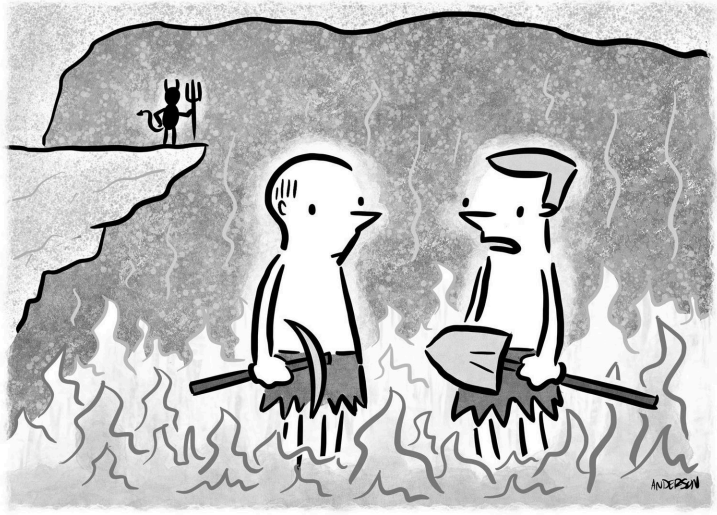
> We've all seen the organic pull from larger, enterprise-sized customers. There's obviously opportunity there, but what is the right way for us to approach it, starting from where we are today?
>
> We considered several options, such as _____ and _____. We decided upon _____ because while it will require [department] to [do something complex and risky], we realized it would be really easy for everyone else because [why it's trivial in other areas]. So, this was the least-risky, highest-chance-for-success way for us to approach this new market, add a new growth area, and learn, and see where it goes from there.
>
> Exciting! And let's remember to give [department] our support and grace as they transform themselves to support this new strategic effort.

Congratulations for hitting Product/Market Fit (p. 335), starting to scale, and now having the "good problem to have" of where to expand next.

Hopefully this will help you make the right decision.

Chapter 55:
Individual efficiency vs administrative efficiency

EFFICIENCY · STANDARDIZATION



"Don't get me wrong, I'm no fan of the heat or the backbreaking toil. It's the corresponding paperwork that gets me."

credit[813]

Everyone has their own favorite note-taking app: Notion versus Google Docs versus Apple Notes versus OneNote versus Obsidian. Dropbox Paper shrieking "what about me?" into the void. I wrote this article in Bear.[814] (No relation.)

From the individual's point of view, the optimal company policy is to allow everyone to use whatever app they like. It's both autonomy and mastery (p. 399). Everyone's comfortable and efficient.

From the team's point of view, however, this is a bad policy. I don't want to learn how pseudo-database-table-thingies work in Notion, but Bear doesn't support sharing notes between people. Efficiency for the individual reduces efficiency for the team.

Beyond the team, there are company-wide administrative concerns. Once we're sharing notes, who has access to those notes? And who decides who has access to which notes? What if company secrets are in those notes? What if customer PII[815] are in those notes? Is note-authentication linked to the corporate identity system, so that when someone leaves the company they automatically no longer have access to the notes? Can the central IT team back up the notes? Can the central security team audit the notes? Since the notes are shared, they probably bounce through a server somewhere; is that secure? Administrative requirements have a negative effect on the efficiency of the team, and certainly on the individual.

While you might be laughing because "this is why big companies are dumb and slow," and indeed startups often win exactly because (p. 295) they don't have to be "dumb" and "slow" like this, it's also why the big company will add more ARR today than the small company will add in the next few months. (A result that is neither dumb nor slow; both the benefits and drawbacks are due to scale (p. 773).)

So, when should a policy optimize for the efficiency and happiness of the individual, and when should it optimize for the team, or the company?

IN SEARCH OF EFFICIENCY

Imagine a scenario in which 10 people each receive a meal. Each meal is unique, and each person has unique food preferences.

A maximally efficient way to *administer* the meals is to dole them out randomly. Nothing to manage or track. This is also *fair*—a desirable quality in policies—in that everyone is (mis)treated identically. Even so, this is obviously suboptimal for the recipients.

Next, everyone looks around the room. Some person `P` sees another person `Q` holding a plate that `P` would prefer to have. Perhaps `Q` is thinking the same thing about `P`! In that case, it's wise for them to trade plates; both are happier. Or the scenario where person `P` would be happier if they traded with `Q`, and `Q` doesn't care either way; they also should trade. Trades won't happen if one person would be less happy; that person would simply refuse the transaction.

If we allow these sorts of trades to proceed until no further trades are possible, the system reaches a state called "Pareto-Optimal."[816] Formally, this is a state where any transaction would result in at least one party being worse off than they currently are.

"Pareto-Efficiency" is not the only kind of efficiency, nor does it necessarily reach a "maximally efficient" state by some measures. Con-

sider the goal of "maximizing total happiness," a kind of utilitarianism, and apply it to the following scenario, where persons `P` and `Q` rate their plates on a 1-10 scale, where 10 is the best:

Person	`P`'s Plate	`Q`'s Plate
`P`	`1`	`10`
`Q`	`5`	`9`

`P` would very much like to trade, but `Q` would not; this is Pareto-Optimal. But if they did trade, total happiness would increase, because currently total happiness is `1 + 9 = 10` whereas trading would result in `10 + 5 = 15`.

So, should we *force* them to trade, pissing off `Q` for the greater good?

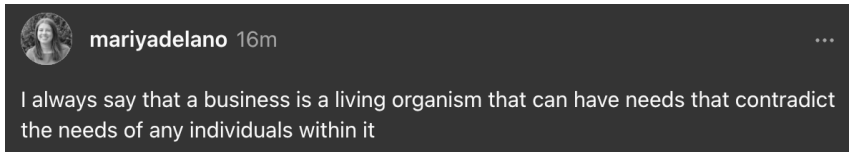
Clearly there won't be a single answer to defining "efficiency" or "fair policy." Nevertheless, with this backdrop as our guide, and taking up practical considerations arising in real companies, there are many actionable things we can do to make policies more fair, and people more efficient, and even more happy.

WHEN TO STANDARDIZE

Under what conditions do the benefits of standardization outweigh the penalties on individual efficiency?

Maximize the individual; satisfy administration

Individual autonomy leads to everything good: The proverbial "win-win" (p. 627) of both happiness and efficiency. Therefore, we should



credit[817]

credit[818]

maximize individual autonomy. Administrative requirements should take precedence only with reason, such as a current problem with harms we can easily identify, or because the law requires it, or because our values dictate it, or because important jobs (like those of the executive team or the security and IT teams) cannot be done without it. Use the Satisficing vs Maximizing Framework (p. 887) to navigate this dynamic.

Minimize the scope of standardization

Most developers agree that code-formatting should be standardized, at least within a single team. One implementation is to require everyone to use the same IDE, which in turn enforces the formatting rules. But that's over-scoping the solution, because many IDEs are capable of enforcing the same formatting rules. The solution is to mandate the *rules*, and allow developers to use any system that enforces those rules. Sure, some IDEs might make that easier than others, but if a developer really wants to use a different editor, and will abide by the rules (perhaps with an external tool and custom automation that the developer maintains), then individual efficiency has been preserved, while the team enjoys the benefits of standardization.

Use standardization in one area to create individual autonomy in other areas

It would easier for the IT department if everyone used the same laptop, with the same operation system, and also used the same smart phone and same tablet. But people have their own smart phones, and would prefer not to carry two. So, as we create the "note-taking policy" mentioned earlier, we could constraint ourselves to consider only those note-taking apps that work well across *all* of Windows, Mac, iOS, and Android. Perhaps we also add the requirement that the note-taking app must have a fantastic web-UI (so that any laptop with any operating system has a good experience), and that it have an at-least-4-star app for both iOS and Android devices. That will narrow the field of possible note-taking systems, but in doing so, we preserve the individual's choice of device. While we're at it, perhaps we have requirements for supporting the visually-impaired and for a wide variety of languages; this additional constraint again increases happiness and effectiveness for individuals.

Standardize on outcomes, not on implementations

Goal-setting and metrics-reporting are common examples of tension between the needs of administration and the needs of operators. The only way for the CEO to keep track of the business is to have a consistent summary of activities, metrics, and how departments and major initiatives are pacing to expectation. But Sales operates very differently from Product; Marketing operates differently from Support. Not just in the obvious ways like which metrics are tracked, but in how work is scheduled, how impact is quantified, and planning cycles. Product might plan tactics every two weeks and strategies annually, yet metrics and goals are reported monthly. Sales typically runs on a monthly cadence and therefore has no problem reporting and reacting monthly. Product doesn't want to ship new things in December; Sales doesn't want any meetings in the final few days of the month. Everyone uses different systems-of-record—Jira vs Zendesk vs Hubspot vs Segment vs Salesforce; there is no natural place for goals and metrics to live. It would definitely be incorrect to force everyone to use a single tool to manage all their work. Therefore, the right solution is to standardize on how goals and metrics are *reported upward*, but explicitly *not* standardize on how each department *operates*. (And to implement a KPI philosophy (p. 645) accordingly.)

Policies should describe explicit benefits for individuals, not just for administrators

No one creates a policy with an explicit intent of causing pain; it happens by accident. If the policy dictates only "what must be done," and not "how it benefits everyone, in different ways," it is likely that the latter wasn't sufficiently considered while the policy was being made. Require policies to have sections that detail how this is beneficial for various parties; if any piece of that section is found wanting, that means our policy isn't good enough yet.

credit[819]

"I *am* multi-tasking! I've done this report three or four times already!"

Look for cases where local optimization creates problems that global optimization can solve

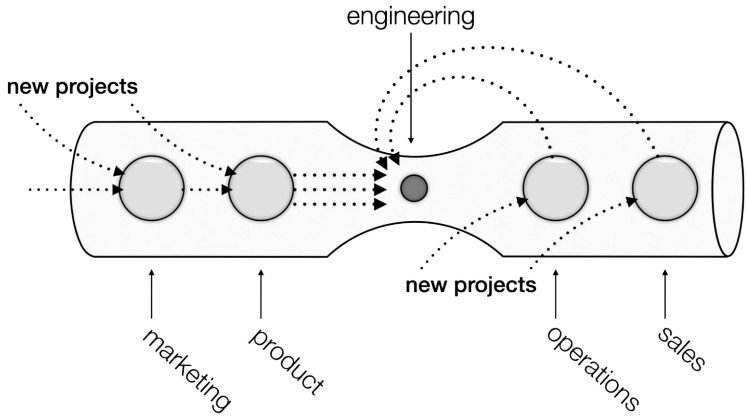
Sometimes, optimizing of one component of a system harms overall system performance. This often happens in marketing funnels. One team is responsible for bringing more traffic to the site, so they optimize and succeed. But this new traffic turns out to be low-quality—perhaps that's why it was so easy to generate—so the "home page → purchase page" conversion rate plummets, harming another team's numbers, and making it harder for them to improve, because now they're wading through garbage traffic rather than improving engagement with quality traffic. This is a moment when global optimization should take precedence over local "efficiency." Another example is in assigning tasks, where people tasked at 90% utilization leads to catastrophic failures (p. 945), to say nothing of unhappiness and burn-out.

Accept sub-optimization in areas that are not the primary bottleneck

In the Theory of Constraints,[820] a system's throughput is limited by a single component (the "bottleneck", Figure 1), and therefore optimizing other components does not increase total throughput (and in fact can ironically decrease total throughput[821]). Among the many techniques[822] for solving the bottleneck is that we should use people who are not part of the bottleneck to unblock or delegate tasks[823] from within the bottleneck, even if these new recruits are far less efficient at executing those tasks, or if the tasks are menial. This is definitionally sub-optimal from the point of view of all teams who aren't the bottleneck, but the result is higher throughput for the entire organization, and therefore it is the right choice. Encourage everyone to participate in diagnosing the problem as well as inventing the solution, so they realize they're working for the greater good, not working for an ignorant organization.

Consider whether local inefficiency is temporary or permanent

All change creates temporary inefficiency, as people and systems acclimate to the change. People often dislike change (p. 1299). It's a well-documented rule of design[825] that any time you change the UX of software, many existing customers will complain simply because it's different. When imposing standardization, we nearly always create temporary inefficiency; assuming the standardization is valuable, we should just accept this cost. But when the inefficiency is permanent, we need correspondingly higher conviction that the standardization is worthwhile.



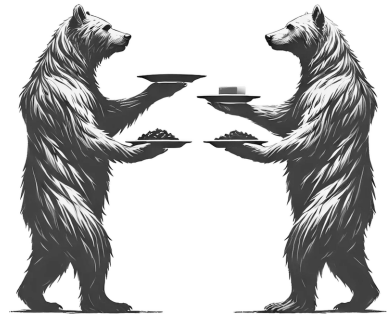
credit[824]

Figure 1: In the Theory of Constraints, yellow activities should be subordinated to help with flow in the red activity.

Document the cases where it's clearly better to (not) standardize

Product-line strategy is shared across teams, so it should be centralized. Communication across the entire company is better as a single report than as five disjointed reports. Conversely, a team's inside jokes only work when they stay inside the team, and no one is harmed when someone wants to use a family photo as their laptop's desktop background. Writing these categories down keeps us honest—not allowing standardization to encroach where is has negative value, and also agreeing that administrative needs should be paramount in certain areas.

Proactively encourage Pareto-Efficiency



In the "meals" example, we claimed that "random assignment" was efficient for management, but also that we could arrive at a better result if people are allowed to trade. Furthermore, trading does not create any problems for management. Therefore, not only should trading be *allowed*, it should be *encouraged*; everyone is better off, even management. Get creative about how people can self-organize within the constraints of the policy. For example: Trade time; some people help another team accomplish one of their goals faster now, and later the reverse happens, all without "management" getting involved. (Also "trading" isn't the only technique.)

A fun real-world example was a motorcycle courier service. Bikers start at Central Dispatch, where they are given packets for delivery and addresses for pickup. They would do a round and return, a few times per day. It's a difficult optimization problem to minimize "total time on the road," e.g. not only to minimize the length of one person's route, but to distribute the packets and addresses intelligently. Frequently the bikers would discover that, for example, one was doing a drop-off just blocks away from where another was doing a pickup.

So they instituted a new system: Every packet for delivery or address for pickup also came with two physical tokens. The way you get paid is by the number of tokens you returned at the end of the day. Every token is worth the same amount. A biker could just perform the assigned route, and make the same amount of money they usually make. But, they could also trade. In particular, a biker could say to another: "If you deliver this package for me, I'll give you one of the two tokens." The giver is happy because they still earn one token for

doing nothing, and the taker is happy because—assuming it's only a small detour—they make more money per hour.

The bikers self-organized far more efficient routes, all made more money per day than they were previously, and the company could handle more clients without hiring more bikers. Everyone wins through self-organized Pareto-Efficiency.

Use global-optimization explicitly, and sparingly

In the "meals" example, we saw that global-optimization can be worse for individuals. It's unclear whether that's "more fair" or "better." And yet, global-optimization *sounds* like the smartest thing to do, and indeed it often is. When it really is smartest, we should be explicit about why that is, what global effect we are maximizing, and why it's so worthwhile for the collective good. If you're going to give me a meal I don't want, at least tell me why the higher purpose (p. 1331) makes it a worthy sacrifice.

All these are variations on the key idea: Individual autonomy should be our paramount goal, and thus our default. But often local optimization does not lead to global optimization, and the latter is what we should all want for our organization.

Since there are many legitimate times when administrative needs *should* supersede the individual, we should always be open to them, but be explicit, be thoughtful, justify the decision, and keep the individual in mind.

Chapter 56:
The "Great" Product Manager, a.k.a. the Impossible Product Manager

DECISIONS · FOUR ROLES
ONE EXCELLENT, ONE GOOD

"What you see here is the culmination of countless suggestions from employees, focus groups, and consultants. Consequently, it's terrible."

credit826

What makes a Product Manager *great*? The prolific Shreyas Doshi* gives us the list of requirements in a tweet-storm:[828]

- Great PMs consistently and singularly improve the company's trajectory.
- Great PMs are masters of the art of blending quantitative and qualitative inputs, as warranted by each individual situation.
- Great PMs become the worldwide experts in their domain. When new to a domain, great PMs bootstrap this process by seeking the counsel of existing worldwide experts.
- Great PMs are diligent about using a variety of user research methods to inform what product to build in the first place.
- Great PMs also listen to what isn't said [by customers] and anticipate where the industry overall is headed when developing their product hypothesis.
- Great PMs know that buy-in isn't enough; you need passion & ownership to build great products. Great PMs facilitate discussions that get the entire team to come up with creative product ideas.
- Great PMs understand task leverage and spend the majority of their time on the highest-leverage tasks for the company.
- Great PMs, in the rare instances of product failure, improve not just their own approach but they also share the lessons learned with the broader company.
- Great PMs are adaptive—they have a wide repertoire [of tools and processes and workflows] that they expertly tweak for each specific team's needs.
- Great PMs are outstanding problem preventers. Great PMs are discerning about which problems to prevent, which problems to solve, and which problems not to solve.

* I highly recommend following @Shreyas,[827] despite the following critique; he is a font of wisdom I have bookmarked and swiped many times.

- Great PMs edit the company's product ethos—they identify the unintended flaws in the principles, fix the flawed parts—and only then follow & espouse it.
- Great PMs know that career ladders are imperfect proxies: they're more fixated on tangible competence & impact than on checking off boxes on the ladder.
- Great PMs also learn through work projects, but they learn a lot more about their craft in their personal time because of their curiosity & passion for self-improvement.
- Great PMs ultimately decide what's best for users & the business.
- Great PMs ensure the product strategy is optimal.
- Great PMs work hard but are rarely overwhelmed.

Haha, that last one is funny. Don't be overwhelmed, but also "learn a lot more about your craft in your personal time."

These recitals always include a little "out," an opportunity for the writer to wriggle out of responsibility for commanding the impossible. Something like, "The final rule is: You can break any rule, if you know what you're doing." In this case, the 31st tweet:

> Naturally, very few PMs are Great.
>
> And I don't know any Great PM who does all of the above, all of the time.
>
> That's because Great PMs know that these ideas should be viewed as signposts, not as commandments.

So you *could* be super-human at two dozen different things. Now what?

I've used the following framework for Product Management at WP Engine to answer this question, so that *as a whole team* we in fact constitute the mythical omnibus "Great PM," but we achieve it with a set of great—but plausible—real human beings.

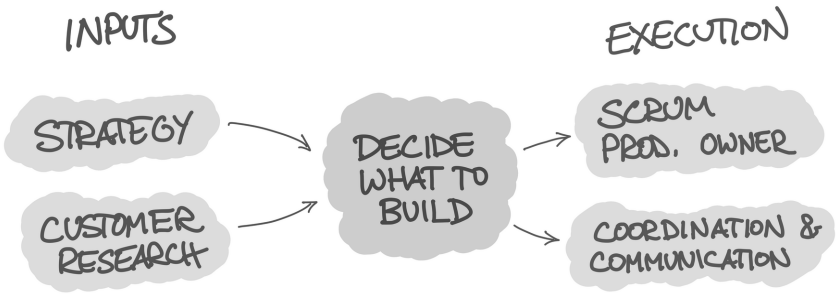


Figure 1

THE PM DECIDES WHAT TO BUILD

The role and scope of "Product Manager" varies between companies, but the role centers on **deciding what to build** (Figure 1).

Looking backward, there are inputs to the decision—the long-term vision and strategy, and an intimate and evolving understanding of the customer; these are part of Product Management, because without them we cannot make good decisions.

Looking forward there's the execution of building and delivering to customers, the complexity of engineering execution together with coordinated efforts by marketing, sales, customer service, and external partners to deliver the entire experience to customers; these consequences of the decision are also part of Product Management.

You could argue that this particular break-down is arbitrary, however in my experience this categorization is especially actionable for hiring, careers, and designing the PM organization, and is compatible with other ways* of decomposing the role.

---

* Product thinkers categorize PM roles in myriad combinations. Lenny Rachistky[829] in a podcast[830] says it's three things: (a) Shape the Product, (b) Ship the Product, (c) Synchronize the people. Later in the same podcast he says it's five skill areas: (1) Execution, (2) Collaboration, (3) Leadership, (4) Vision + Strategy, (5) User

THE FOUR ROLES OF PRODUCT MANAGEMENT

Strategist
Analyzing, crafting, communicating, and updating the answer to the question: How do we apply our durable advantages to the opportunities in the market, to win? What do we look like in the success-case, years from now? What are the most important challenges to achieving that? Specifically what must we do in the coming years to overcome those challenges? Competitive analysis, market analysis, customer analysis. Determining the few, key personas to focus on. Distilling noisy data into clear identifications of internal strengths, external challenges, and market opportunities (p. 71). Positioning the product to maximize value and perceived value, and thus price (p. 165) and willingness-to-pay (p. 275). Selecting the right high-level metrics (p. 645) to drive the right long-term results, while also acknowledging that not all important things are numbers, and that revenue and retention are outputs of great strategy plus great execution, not themselves a strategy or even an operational goal. Having the fortitude to make decisions that will affect us for years, and cause us to say "no" to 90% of our ideas, so we say "yes" to exactly the right ideas.

Customer Whisperer
Not just listening to customers, *hearing* them, understanding their underlying needs, motivations, even their emotions. Not order-taking the features they know how to name, but rooting out the things (p. 239) they worry about so much they would pay to decrease the worry, or figuring out the result they need to get to earn a promotion or achieve their ultimate goal (p. 259), as opposed to just "features." More than figuring it out, *wanting* to figure it out. Having the "taste" to sense what features would be especially useful even in the absence of clear quantitative data, or the nuance that separates a serviceable UX from a delightful one that causes an NPS-10 customer to actually "Promote" in

Research. In another blog post he lists the ten jobs of Product Management.[831] You'll find the four areas below align with any of these break-downs; while all paths arrive at the same destination, I find this one particularly useful in working with real-world people.

real life. Translating these insights into tangible feature ideas or use-cases of things to build that make the customer a hero.

Scrum Product Owner
An unfortunately ambiguous phrase for the uninitiated, the Scrum "Product Owner" owns the work-backlog, and works with engineering to execute it. Owning the work from start to finish, breaking down epics into stories, prioritizing and scheduling (p. 221) innovations for customers against engineering demands against requests from other teams (p. 711) against the broader needs of the company. Working tightly with engineering not just to shake them upside down until estimates fall out of their pockets, but working together to solve the puzzles of "most value (whatever that means) in the least time (or risk)," together finding the 80% solution that will take 20% of the time, so that we deliver quickly and learn quickly. Using the tools of story points and velocity and retros for team-wide constructive introspection and improvement, fulfilling the responsible aspect of being "self-managed." Stretching to maximize how much we deliver, but practical and celebratory so that constant striving doesn't turn into death-march burn-out. Being in command (p. 413) of the product and team.

Orchestrator
Delivering the "Whole Product,"[832] as Geoffrey Moore defines it, means not delivering working code only, but also enabling others: Marketing to grab attention and spark curiosity, sales to convert potential energy (leads) into kinetic energy (sales), customer service to shepherd the customer through good times and bad, enabling the Marcomm teams for PR and events, and external integrators and consultants who operate on behalf of clients. Tracking and communicating status, the good news and the challenges, celebrating the wins and facing the challenges with crisp articulation, calming stakeholders in the knowledge that the team is fully aware of what they need to overcome, and thus in command even in dire situations. Running great meetings—starting with a clear goal and how the discussion will support the strategy, tight agendas so participants come prepared and use synchronous time wisely, with decisions made and recorded. Engaging the whole company on occasion, with inspiring presentations at all-hands meetings so everyone is excited about we're accomplishing together, and so that the teams working directly on the product feel proud, and feel seen.

"Fire good. All like fire. But sometimes fire not good. That why Og make exciting new product. Og call it 'Bucket of Water.'"

credit[833]

The conclusion is not that “a Great PM must be the master of all this.”

In fact, exactly the opposite.

ONE EXCELLENT, ONE GOOD

In my experience, echoed by a few people I’ve chatted with, there is a general rule of thumb vis-a-vis the categories of work above:

A “Great PM” is
***excellent** in one area,*
***good** in at least one other,*
and doesn’t have time for more than two.

Using "Great PM" language, you should strive to be "Great," as superlative as possible, in one of those four areas, and pretty darn good at a second.

I find this particular categorization of four job-areas aligns well to strengths and abilities of real people. Whatever you ascribe to nature or nurture, instinct or experience, for each of those job-areas I can immediately recall specific people who are excellent at nearly all components within a single area, but variously good or lacking in components of a second, and then completely the wrong temperament or lack of desire to excel in others. These four "buckets" seem to me to be natural divisions of ability and proclivity.

I don't believe in magical unicorns who are excellent in all four areas.* But suppose they do exist, and you hired them. Would it be physically possible to execute well in all four areas? It's easy to see that it's impossible, if you add up the time it takes to do great work.

Strategy is on-going—markets change, competitors change, customers change, new data comes to light (usually only after expending considerable time acquiring it), and whenever you're "pencils down" on one iteration of a strategy, there are always eight more things you still haven't figured out yet. Customer discovery takes a long time—scheduling and running interviews, compiling results, mapping what you heard, prioritizing activities or opportunities or use-cases (depending on your frameworks), trying to (in)validate assumptions with data. Being a Product Owner takes at least twenty hours per week to write great stories and run Scrum ceremonies. Anyone who has program-managed multiple teams across multiple departments knows that scheduling and organizing and cajoling and status-updating and data-gathering and meeting-preparation takes a lot of time. Even with sensational skills, no one has time to do it all (with excellence).

---

* Perhaps there are, but the probability that you'll find one, or hire one, or build a whole organization of them, is near-nil, although you will surely find a lot of candidates who imagine themselves to be one.

> *"We are not going to breed a new race of super-humans; we will have to run our organizations with the humans we have."*
>
> *—Peter Drucker*

While one person cannot be all things to all people, across the *entire* product team, you *do* need excellence in all four areas. Thus the manager of a Product organization isn't simply a "manager," but rather an "organization designer," choreographing this outcome through intentional hiring.

You'll often find these roles in titles other than "Product Manager." Communications and collaboration is often done by a Program Manager—a role where if you think it's not useful, it might be because you haven't worked with someone who is great.[834] Marketing, sales, and support enablement is often done by Product Marketing (i.e. somewhere under the CMO rather than somewhere under the CTO). With highly technical back-end teams with little or no customer exposure, a sociable engineer or architect can be the perfect Product Owner. Mature organizations might have UX Research teams who are expert in customer analysis—building data-based personas, crafting interviews, compiling results, and pooling insights across multiple product teams who happen to share customers.

As the organization designer, your job is to "assemble excellence" in all areas. Even incremental improvements here can be impactful, because so often organizations have excellence *nowhere*. 95%[835] of employees complain they don't understand the strategy, or don't know their role in it; founders-in-PM-roles often run on instinct more than systematic customer analysis; engineering teams often complain of underspecified or under-explained stories; engineers are often surprised at the gulf between what they know to be true about the product and what Sales claims on the phone, just as customer service folks

are often surprised at the gulf between what they know to be true about the product, and what engineering thinks is "a job well done."

Rather than unicorn-hunting, take this more practical route in your career, or in designing your team.

Chapter 57:
In its emptiness, there is the function of a startup

"I didn't say I was unfulfilled. I said I was hollow."

Calm your frenetic TL;DR twitchy-Twitter-brain for five minutes, and let some Chinese philosophy wash over you.

> Thirty spokes join in one hub
> In its emptiness, there is the function of a vehicle
>
> Mix clay to create a container
> In its emptiness, there is the function of a container
>
> Cut open doors and windows to create a room
> In its emptiness, there is the function of a room
>
> Therefore, that which exists is used to create benefit
> That which is empty is used to create functionality
>
> —Lao Tzu, Chapter 11, Tao Te Ching, ~600 BCE, translation[837] by Derek Lin

An ancient brainteaser asks: If a wooden ship sails around the world, but along the way each board rots and is replaced by new boards, so that by the end of its journey every board has been replaced, is it still the same ship? Did "it" really go around the world?

Lao Tzu suggests the answer is "yes," because the functionality of the ship—the "emptiness" created by those boards—is the essence of the ship.

What about yourself? Your boards are being replaced every second—trillions of cells splitting, growing, dying. Are you the same person you were ten years ago? Physically, not really. In DNA yes (mostly), but identical twins share DNA and are not the same person. Can you articulate precisely what resides in the emptiness created by your body, or by the electro-chemical pandemonium sloshing within your skull?

What about a startup? It's not a person[838] but composed of people. Is a company of 2 people and 150 customers the "same company" as with 150 people and 25,000 customers? In tangible ways it's not the same, but in its emptiness lies the essence and purpose of this motley crew of human beings.

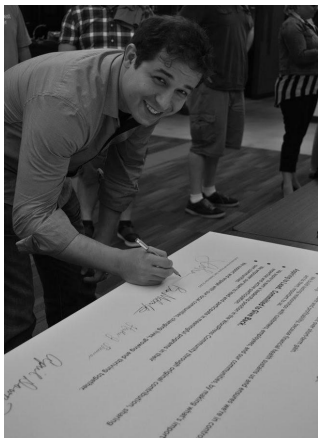
So the interesting question for a startup is: What is constant? What *doesn't* change as we grow, and thus what binds us together in this journey? If we "go around the world," changing out every board, what has been conveyed?

Can't be about product features, they change, grow, and die like cells. Can't be about marketing prose, that's molded by A/B testing into market-resonance. Can't be people—the team's composition changes, and those who stay for the entire voyage themselves change. Can't be the metrics (p. 645), they'll change by orders of magnitude. Can't be the logo, or even the name. Can't be the office space—Lao Tzu already made that clear, as did COVID.

So, what doesn't change? Whatever it is, **that is what defines the startup and its purpose**, and is the answer to the all-important question: Why should anyone—employees and customers alike—join the founder on this journey, sharing this very personal responsibility?

At WP Engine we call these constants our "values"—our inviolable constitution. When we need to make a tough decision, our values determine the outcome. For example, if in some choice we must either lose money or do what's right for the customer, we will do what's right for the customer. Or if we have to decide whether to fire someone who is exceptionally skillful at his job but is dishonest or works for himself instead of for the team, we will always fire that person.

In fact, a clear agreement of core values **is how you empower everyone in the company to act like an owner.** Can someone in their first week of tech support decide whether it's OK to give a customer a large refund or spend extra hours to go above and beyond to fix something? If you have a framework of values which clearly points to that as the right course of action, then someone can make that decision without approval-chains and meetings and debates. Isn't that the definition of empowerment? And doesn't that sound like part of the solution to "scale?"



Signing the WP Engine values

This is why it's so important to encode your company's values on paper. For us, we literally put them on the wall and we all signed it like the US Declaration of Independence. You have to ensure everyone not only agrees to them as a condition of employment, committing them to memory due to external requirement, but because in their bones they already embody those values, and therefore will naturally continue propagating them to those around them and those who follow.

"Company Culture" is the output of living your values. It's never perfect, either as individuals or an organization, and at any given time you're falling short at some aspects of the culture, aspirationally trying to live up to your own expectations. That's healthy.

The three companies I founded before WP Engine were successful enough, but I never thought about culture or values. That was a mistake. Today two of those companies still exist, and after nearly two decades have continued to grow, and yet, without a higher purpose, a truly important mission for our customers and dedication to our employees, **maybe who cares?**

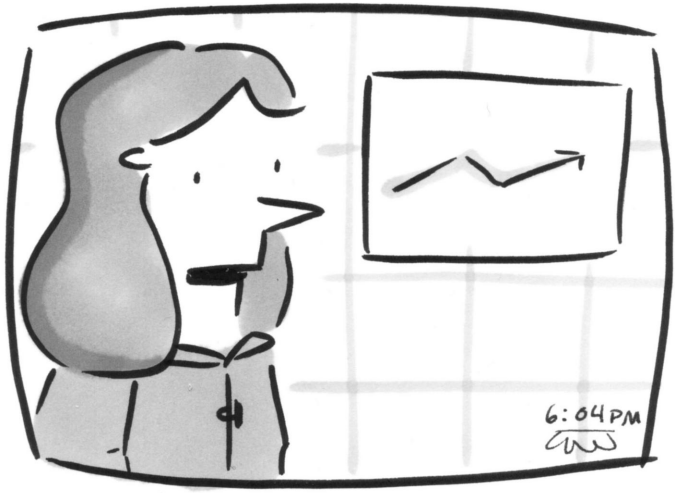
Why so dismissive of that success? Because what sort of jobs did that create? Were people happy and empowered? Mentoring and being mentored? Did they build a career they're proud of? Did they feel that this time in their life was not wasted (p. 861)?

I don't actually know, how sad is that? **Financial success is the means, not the ends.** Without financial success the company isn't sustainable, so that component of success is absolutely necessary. But "money doesn't make you happy," and it certainly doesn't make other people happy, or lead to a fulfilling, meaningful life (p. 399).

What does? Answer that question, revealed by your culture in the emptiness created by your values, and you've truly done something important with your life, and for others' lives.

What is your emptiness?

Chapter 58:
No wait, of course THAT is the single most important SaaS metric



"Reaction to the news was mixed, largely because we asked more than one person."

The single most important SaaS metric is retention, because cancellations[840] indicate lack of Product/Market Fit (p. 335), no matter the cause (price, features, severity of need, duration of need). If it cannot be fixed, it means the business is a failure even if other metrics are stellar, because it means customers are rejecting the product. Once you're scaling (i.e. past Product/Market Fit and into tens of millions in revenue), you realize that cancellation scales exponentially with total customers—whereas new-customer growth scales linearly with sales and marketing costs—which means cancellation necessarily and always outgrows inbound activity, thus growth ceases. So, retention is most-important, regardless of stage.

The single most important SaaS metric is top-line growth, because high revenue growth implies many other (even un-measurable) things are going well, such as Product/Market Fit, ability to get to customers, ability to retain customers, and proving the large size of the addressable—no really, we're actually addressing it! (p. 71)—market. High growth is the only way to build a large tech company, because tech market dynamics follow a "winner take all" power law,[841] and the only way to be the biggest is to grow the fastest. Fast growth proves every other piece of the company is healthy, or at least healthy enough not to be fatal. High-growth companies have all the options—staying independent, raising money, selling. It's also by far the greatest determinant of equity value, because it maximizes both revenue and the revenue-multiple.

The single most important SaaS metric is net-profit growth, because the days where equity value is determined almost completely by top-line growth died in 2015,[842] and now both public and private markets are rewarding not just growth, but profitable growth. Which is better anyway, because it means we're building sustainable, value-creating companies, not just companies that grow like a virus (p. 1331),

without purpose and possibly killing the host. The unicorns are dying not for of lack of revenue, but for lack of profit.[843]

The single most important SaaS metric is sales efficiency, because if it's expensive to acquire new revenue, it means the product is being stuffed into the market rather than embraced by it (p. 1469), which means you're artificially growing rather than having found real Product/Market Fit and longevity. It will be too hard to grow at scale because merely replacing natural cancellation is expensive (p. 1361), thus your maximum size is capped. The expense means it's too hard to be profitable in the long run, because too much of a customer's lifetime revenue is spent in costs before they even write the first check. It means you're "buying customers" rather than building a product people organically want, enjoy, and talk about (p. 275), which in turn calls into question the fundamental value of the product.

The single most important SaaS metric is LTV, because if you look at the component metrics individually you never get a complete picture. High growth is great, but not if margins are so low there won't be a profitable business at scale. High retention is great, but not if MRR is so low (p. 515) that sizable revenue requires too many customers. When you optimize one number in LTV, often the others change in contrary directions. For example, you can increase price which increases MRR but also increases cancellations as people can't afford it. Only by combining these metrics can you get a clear and true picture of the health of the business model. Some people say LTV isn't valid (p. 1355) because the future is unpredictable (p. 193), but that's exactly why it's so important to try.

The single most important SaaS metric is the Rule of 40[844], because either your company needs to be growing fast (in which case being unprofitable now creates long-term value), or your company needs to be very profitable (where generated "value" is literally cash coming out of the business, over many years). The "Rule of 40" achieves this—the

sum of YoY revenue growth rate and bottom-line profit margin—the "super-metric" that covers both the "growth" and "profit" phases of a successful company's lifespan.

The single most important SaaS metric is NPS (customer satisfaction), because this is the ultimate measure of value-creation, which in turn impacts every other important metric. It impacts new growth because only high-NPS customers will propagate word-of-mouth marketing—the least expensive form of marketing and one of the few that continues to scale with the business. It impacts retention because high-NPS implies they'll stay. It even impacts objectively important things which don't have numeric metrics, such as brand quality and Product relevancy and customers who love you even when you screw up (p. 1559). It covers not only Product Management (p. 817) but Tech Support (p. 1503) and every other function. The customer is the fundamental unit of the company, and NPS measures customer health.

All of these statements are true, for some companies, at some times, in some stages, in some industries, in some markets. But the larger point is that companies are more complex than "a single most important metric."

The purpose of a metric is to be a tool in service of (p. 645) your goals, timeline, size, circumstance, even philosophy, not a master you that you thoughtlessly obey.

Focussing on just one thing is valuable because it creates focus, especially when a company is young and there isn't enough time to maximize multiple goals. Larger companies like WP Engine can have three or four, but still not many.

Your job is to figure out what's most important *right now*, what's on fire, what's most important for getting the company to its next mile-

stone, lean into what's working well, and then *simplify* and *clarify* the few goals and metrics that your entire company should align on. Like this (p. 1065).

Chapter 59:
The Lindy Effect on startup potential

> *"The first 90% of the code accounts for the first 90% of the development time. The remaining 10% of the code accounts for the other 90% of the time."*
>
> *–Tom Cargill, Bell Labs*

However long it has taken us to get a project "close to done," it will probably take that much time again to *really* be done. **It's funny because it's true.**

Why is it true? Here's a novel way to frame it: When you're exploring something new, where the terminus is unknown, you never know how far along the path you are. **On average, however, you're halfway there.** This is due to the very definition of "average"—you'll spend 50% your time before the half-way point, and 50% after.

credit[845]

"No, it's too late for the oversight committee, send it to the hindsight committee."

This general rule is called the Lindy Effect:[846] For certain non-perishable things (like technology, companies, books, and ideas), the expected lifespan is twice its current age.

Rewriting the same rule with different language allows us to apply this rule to startup growth:

However large you've gotten, you can probably double it.

If you've gotten 10 customers you can probably get another 10 in a similar way. Will you ever get 2000? I hope so, but *most* companies that do get 10 *never* get 2000. Putting it another way, doubling the size of the company always sounds plausible, because you've done it once, so you can probably do it again, even faster this time. But 10x or 100x is not obvious at all.

One way to understand why 2x is plausible but 20x requires innovation, is to observe that the actions that got you your first 10 customers are probably not sufficient for generating 100, even though

they're probably sufficient for getting another 10. You might have scratched and clawed inside your social network to get the first 10, but that doesn't scale to 100, and just because you were successful at convincing customers one-at-a-time to convert through hour-long Zoom calls doesn't mean you can convert 100 no-sales-touch customers via Google ads.

Or you might have gotten to 1000 customers through one marketing channel. Although surely that same channel can produce another 1000, it's unlikely that there is 10x the inventory inside that one channel (p. 115) to get you to 10,000 customers. Thus you'll need to make other channels work, which is not easy to accomplish, as anyone who's gone through that challenge can attest. In fact, to achieve 10x you'll probably need *multiple* new channels. Yikes! Possible? Sure. Likely? No, or at least, not with the confidence you have in 2x.

So the "startup growth" version of this rule is:

You can probably double your size by doing what you're already doing, but 10x will require innovation.

Figure 1 shows an example.

Or, instead of innovation, time and luck. Specifically, waiting a long time for the existing mechanism to keep working, and lucky that nothing stops that slow but predictable growth: the channel doesn't saturate and get worse, new competitors don't arise, market conditions don't make the product less desirable, the economy doesn't slow, and so on. Again, *possible*, but in the fast-moving world of tech, *unlikely*.

Indeed, at some point 10x is strictly impossible. At the high end, you hit market size limits (Facebook has 2B users; there aren't 20B humans) or run out of marketing channels for acquisition (GoDaddy's customer growth rate is 13%/yr since 2009;[848] at that rate it will take 18 years for them to 10x, assuming market size and conditions would even allow them to sustain that). At the low end, maybe there really isn't a market, or the market really doesn't want that product, at least

Dmytro Krasun
@DmytroKrasun

New milestone 🙌🏻

200 paying customers with a monthly ARPU of $37.

But my goal is 300 with an ARPU of $34. I crawl to $10K MRR however I can.

But I can't imagine doing the same to reach, for example, $100K MRR. It will require something new, something I haven't done before.

credit[847]

Figure 1

not at a profitable price, so you can squeeze out some early sales but it can't get substantially bigger.

Editor's Note: This article was written in 2014, so now in 2024 we can actually look back 10 years and see whether my GoDaddy prediction was correct. Annual revenue in 2014 was just under $1B, and in 2024 it is about $4B, so indeed they did not 10x—didn't even 5x—over a ten-year period. Also, GoDaddy expanded into several additional product lines; had they "done what we've always done," the growth would have been even lower. Their growth since 2014 (Figure 2).

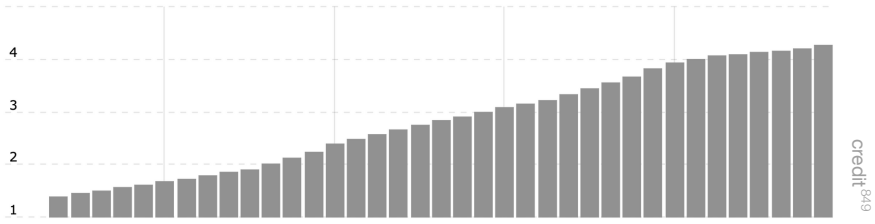


Figure 2: GoDaddy's annual revenue, in billions of USD, 2014-2024

You get 2x by assumption but 10x you have to prove. If you're in the business of raising money, you actually have to prove it. But how do you prove something that we just agreed was unlikely?

One answer is having *very fast* growth. If you're doubling every six months, you clearly have line-of-sight for more growth than just 2x. Your trajectory proves intense market demand is getting coupled with an ability to find and service it. If the underlying market is large and/or growing, you have a good case that 10x is already within reach, and innovation could potentially get you 100x or 1000x. And indeed, the companies who have shown that sort of growth at interesting size * have indeed shown 100x or 1000x size thereafter.**

This is why investors (and founders) wishing to build enormous companies are so fixated on *hyper*-growth.*** It's the only way to have even the potential of building something enormous.

Another way to break the Lindy Effect is to change something substantial. I think of this as expanding into an adjacency (p. 793). As explained in that article, this means keep one foot planted in the areas you're already strong, and expanding into something new, risky, but

* "Interesting size" doesn't mean going from 10 customers to 20 in six months and then being proud of your "rapid growth." Going from 1000 to 2000 in six months is more like it.

** With 10 years of hindsight, our company WP Engine was another one of these which experienced hyper-growth (p. 335) since hitting $1M ARR, and achieved 100x that six years later.

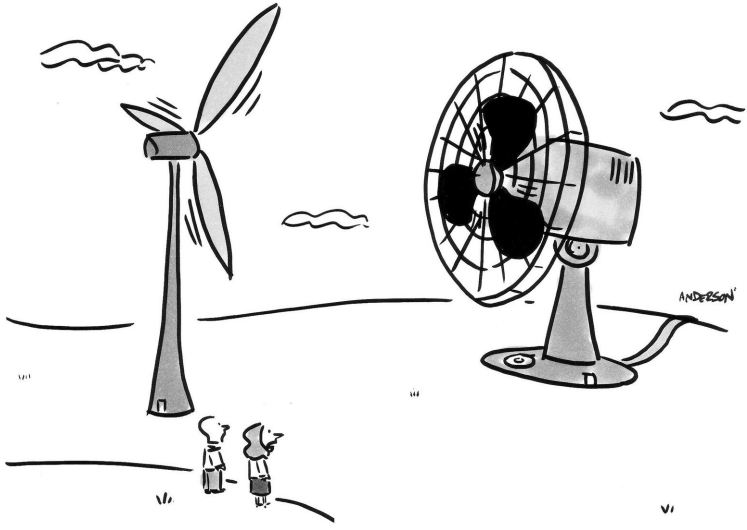
*** And "exponential growth," though that is a misnomer (p. 115).

with much higher upside than incremental improvement on the existing business.

Of course not everyone cares about building something that services a million customers, or cares to expand into other areas for the sake of growth. Nor should they.

Even so, the Lindy Effect is helpful understanding what will be easy, and what will require something new.

Chapter 60:
The “Convergent” theory of finding truth in darkness



“I dunno, kind of defeats the purpose, doesn’t it?”

How do you know if your startup idea is a good one? Even after twenty systematic customer interviews (p. 239)? So often, people say nice things to your face, but really they're not buying.

How do you know when to give up and try a different idea?

The usual answers: It's a balance (p. 589). Trust your gut. But your gut is wrong so trust data. But you don't have data so trust your gut. Don't give up just because it's hard (p. 159). If it's hard it's not a good fit (p. 9). Don't listen to the haters.[851] Haters have something to teach us (p. 751).

Follow the formula (p. 239). The formula says there's no one formula. The formula is to ask "the right" questions. The formula doesn't know what to do with the answers. Everyone is a unique snowflake. "It depends."

Until recently I haven't had a good way to explain my idea of the answer. But recently I was rereading Richard Feynman's Lectures on Physics[852] and, in one of those brief flashes of comprehension that comes when your mind is active yet wandering, I stumbled across my explanation.

Fluid dynamics are famously complex and incalculable. Consider the drag forces on this Harrier jet (Figure 1)—swirling currents pulling it backward as it crashes through the air, sluicing through curved surfaces, jarred by every seam and rivet, twisting and colliding with itself. It's so chaotic, even modern simulators can't model it perfectly, so we resort to wind tunnels, where we ask the universe to just tell us the answer.

Given this complexity and chaos, it is astounding that the force of drag on an airplane can be approximated as being simply proportional to the square of its velocity. That is:

$$F_{\text{drag}} = Cv^2$$

How could such a simple formula summarize such a fantastically complex process?

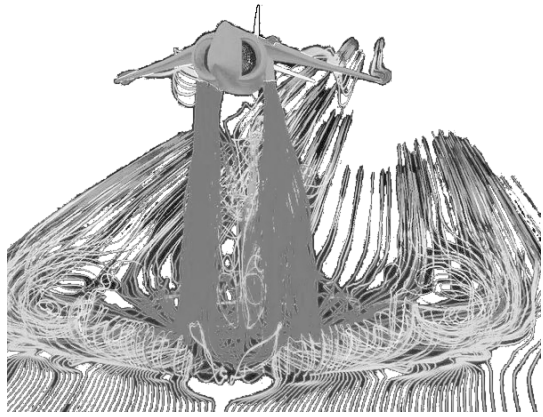


Figure 1: Airflow around a Harrier jet

Humans love this sort of thing—emergent simplicity from complex chaos. There's beauty in its brevity and power in its utility. We love it so much, the urge is in constant overdrive, and we see patterns and meaning even when there are none (p. 1487).

We're tempted, therefore, to call $F_{\text{drag}} = Cv^2$ a "law"—a rule by which a mere human brain can get a handle on a phenomenon too complex for the fastest supercomputers.

But it turns out it's not a law at all. It's not power, it's a tenuous coincidence, and not one of great utility.

How do we know this?

Because as soon as we try to understand similar situations using this law, it breaks down. If the airplane is flying slowly, it becomes completely inaccurate. If the airplane flies very fast it's wrong again. If we make small perturbations on the wing, the constant C can change dramatically. If you physically remove one of the wings, that changes the drag on the remaining wing.

There are forces inside and outside the aircraft, hidden to a casual observer, uncaptured by simple formulas.

Where is all this in our simple "law?" Nowhere. If we step outside our little safe space, the law crumbles. It's not a *fundamental* law, because it does not predict what happens in novel situations.

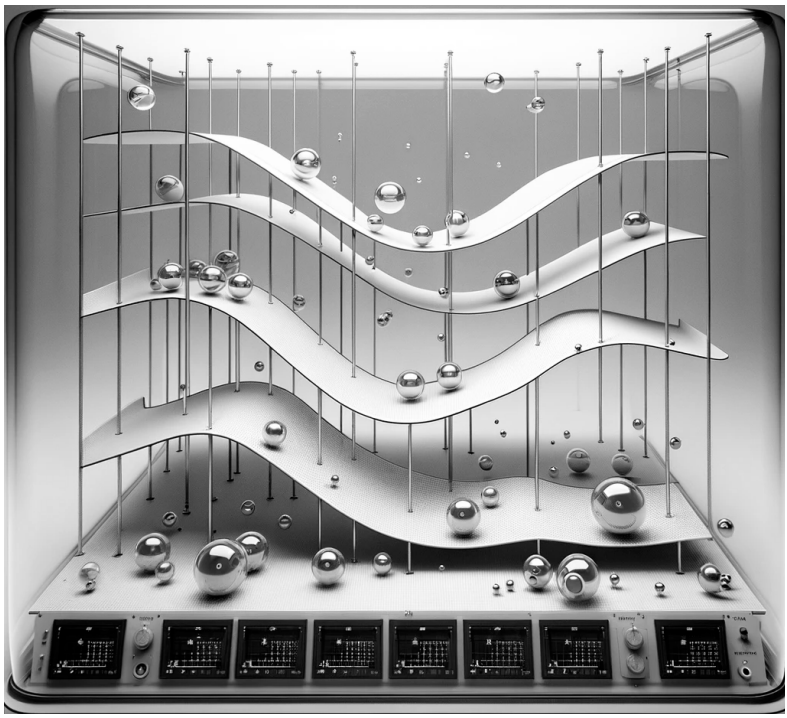
Contrast this to another so-called law—Conservation of Energy—which states that the total energy of a closed system remains constant over time. So if a ball falls in a gravitational field, it loses potential energy (height) while gaining kinetic energy (speed), such that the total energy never changes.

Is this a true "law?" How can we tell?

We can make a complex series of ramps inside a vacuum, starting a ball at different heights and positions and letting it roll down and up and around, measuring the velocity the whole time. We find that the ball's speed everywhere exactly ensures the two energies remain equal, regardless of the configuration of the ramp. This feels powerful—in arbitrary configurations, the law accurately predicts the result.

With a real ball and a real ramp, friction slows the ball, thereby reducing total energy and therefore a violation of the law! Ah, but we realize that "heat" is also energy—something we can measure and convert into other forms of energy—and when we measure the increase in heat in our ball-and-ramp experiment we find that the energy due to heat exactly replaces the energy lost as the ball slows, and again our law is proven correct. In fact, our law predicts *how much* heat, and we find exactly that amount, so now the law has just predicted the existence of new *kinds* of energy, and did so with accuracy, which is even more impressive!

Then from other experiments we learn that matter is in fact composed of gargantuan quantities of tiny objects (molecules, atoms), moving and colliding and vibrating. That suggests a different definition of heat itself—that it's not a "new" form of energy at all, but rather the total kinetic energy from jiggling particles! Under this hypothesis we can make definite predictions about how much energy heat contains, how heat and particle density and pressure would change in a gas under various conditions, and so on, all on the sole basis that



energy must be conserved, and in fact all those predictions are again accurate.

Even in the modern era with Relativity bending and weaving time and space, and Quantum Mechanics so strange that Feynman himself said that no one really understands it, still the conservation of energy has always been found to be perfectly correct.

With the drag-force equation, the deeper we dug the more we discovered that the "law" doesn't encompass much truth; with Conservation of Energy, the closer we look the more powerful the law becomes, the more applications we have for it, the more accurate predictions it makes, and *that* is the characteristic of a bona fide Fundamental Law of the Universe.

Truth in startups converges or diverges in the same way.

Specifically, before I validated the ideas behind WP Engine, I validated another idea for a startup (p. 239). The key thing to notice is that during my customer development, everyone said "That's a great idea, you should do it!"

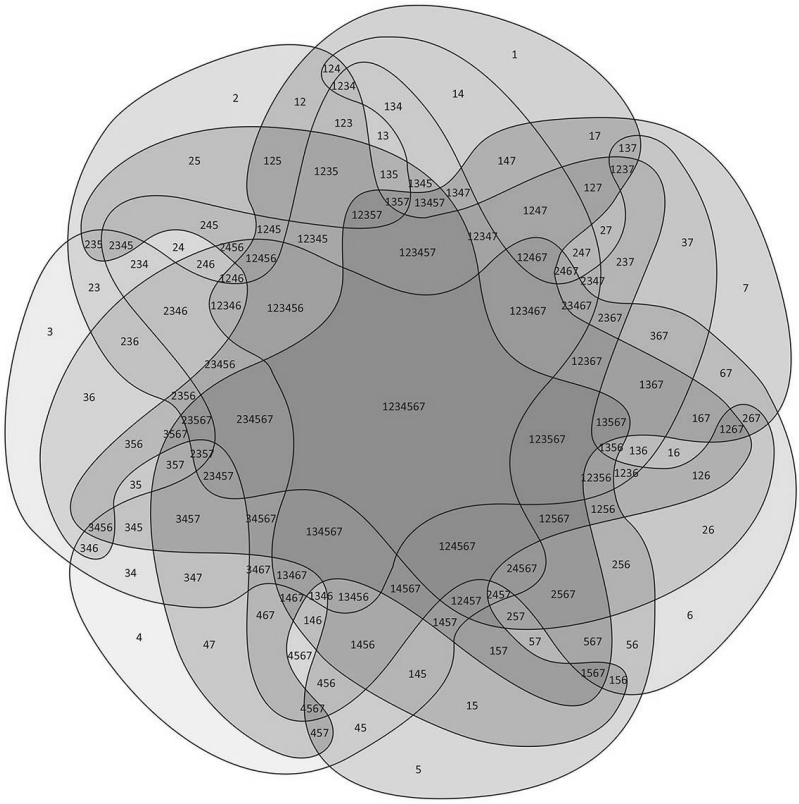
Everyone.

Except, as I dug in with each person, the "truth" started diverging. One said I should target enterprises, charging $1000/mo and selling through consultants. One said I should make it freemium and figure out how to make money converting 5% to $5/mo. Another said charge a minimum of $50/mo to cut out the moochers who email support but don't pay for stuff. Another said the small-to-mid-sized business market[853] is the untapped niche. One said I should use it to measure online ads and forget about measuring leads; the next said I should use it to feed leads into Salesforce and forget about measuring online ads; the next said I should use it to reveal general marketing efficacy and not try to close leads.

Like the airplane law, I had discovered something intriguing, even exciting, but not something fundamental, not something with clear steps forward, not a Venn Diagram of ideas creating a large, dark area filled with perfect customers (p. 317), but rather a blotchy Venn Diagram with twenty lobes of dissonance and no discernible center (Figure 2).

But my experience vetting WP Engine was convergent. The more people I spoke with, the more agreement there was over the pain they had, whether my solution was an acceptable, and the amount they were willing to spend. $50/mo to make a WordPress site fast, scalable, secure, and when tech support answers the phone, they should be knowledgeable about WordPress. Kick in a staging area and backups and it's a done deal. Thirty of forty people agreed to sign up during their interview. (Twenty of the thirty later did, and we launched with a total of thirty customers. More of the story is here (p. 9).)

I had found the startup equivalent of a fundamental law—not an immutable physical law of course but something that behaves like



credit 854

Figure 2: Branko Grünbaum’s[855] 7-color Venn diagram, with regions for every combination.

truth—where multiple areas of inquisition lead to a common destination instead of leading to different planets.

Of course there’ll be no rubric to determine whether an idea is tenable or whether the situation is so bleak that you should give up. But in my experience this feeling of convergence or divergence is very strong if you’re being introspective and honest with yourself (p. 657).

Your hardest battle is indeed with yourself, as you’re constantly tempted to bias the evidence in the most convenient direction (validation), and your fear of figuring out that your pet idea, while un-

credit 856

deniably cool, is not a business, in that other people are not going to give you money for it.

Just remember how expensive it is to remain ignorant. You will meet the same, bitter end, only after a significantly larger investment (p. 867) in time, money, heart, and reputation.

Chapter 61:
On the (un?)importance of design

We recently* underwent a Cinderella-like transformation: A total redesign of the WP Engine[857] website from despicable steaming pile of hideousness to a designed, thematic—dare I say artistic?—sleek new look.

Does it matter?

It must have mattered. Look how bad it used to be. Not only were the pages ugly, they were peppered with database errors and CSS blowups (Figure 1).

Just look at us now, sporting a grayscale 1950s automotive motif playing off the "engine" concept using the latest in CSS3/HTML5 trickery (Figure 2).

Customers emailed us saying "Thank God you fixed that horrible website. I was embarrassed when referring you to my clients."

But hold on. **They were still customers. And they still were referring us to clients.** So I wondered, did it *really* matter?

Modern Lean Startup (p. 1425) theory blares out from the red-tiled rooftops of Stanford: *Seek the Data and Ye Shall Find!*

---

* Editor's note: This was written in 2011.

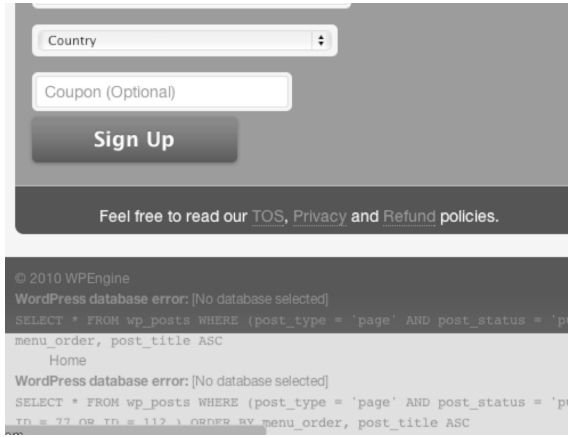


Figure 1: Database errors shooting out the footer of the home page!

Figure 2: When customers have a "crush" on you, that's not just Utility, that's Love (p. 275).

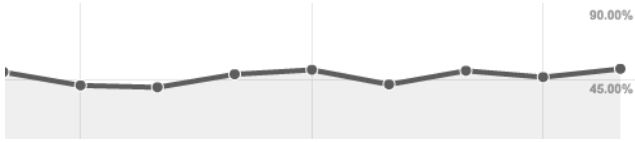


Figure 3: Home page bounce-rate before and after website redesign

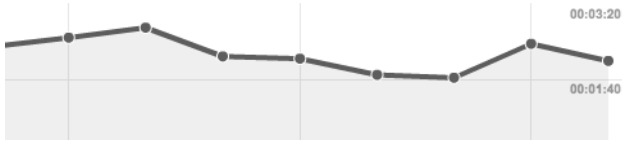


Figure 4: Average time-on-site before and after website redesign

First the bounce-rate. If our website design was repulsive—literally—the bounce rate should go down. Here's the data (Figure 3).

Can you see at what point in time we changed design? No? Hmm.

Let's look at average time-on-site (Figure 4).

No change.

But this is all superficial—what **Really Matters** is the **Conversion Rate**: are more or fewer people signing up each week (Figure 5).

Hmm. **Looks like the data are objectively saying "design doesn't matter."**

But as much as I respect Lean Startup theory, objective measurements aren't the only things that matter. Those customer emails matter too.

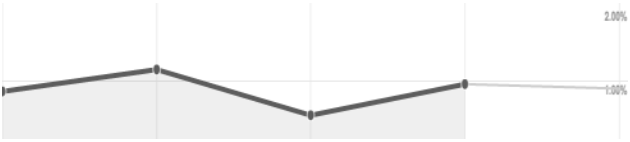


Figure 5: Sales conversion rate before and after website redesign

Maybe the most interesting change was in our own team. I heard things like "I'm *soooo* glad we fixed the site. I was really embarrassed by it." That matters.

The other day we landed a large customer who said they could tell from our website that among our competitors we're more mature and ready to handle a bigger customer like them. I can tell you—objectively—that we were not particularly mature, and although I have a list of reasons why "we're better,"[858] the truth is that particular customer would probably also be on fine on a competitor. Was it the design that gave us that edge? Could be. Didn't hurt, anyway.

Still, the more I look at the importance of design in the startups in my little career, the less it seems to matter. I've chronicled the eye-melting design (p. 193) that punished potential customers in the evolution of the Smart Bear website, and yet with all that cringe-worthiness, that was a company that doubled in revenue *and profit* every year like clockwork for half a decade—a stat any startup would be proud to match. This doesn't prove design *doesn't* matter, but it does suggest design may not be the deciding factor.

An even more extreme example comes from my second company ITWatchDogs.[859] I displayed its old homepage at the magnificent Webstock design conference[860] in Wellington earlier this year; the crowd whooped at our violent assault on the visual arts, complete with calliope menubar colors, two broken images tag above the fold, and a layout model that could be seen as a "grid" only after consuming a pillowcase of mushrooms (Figure 6).

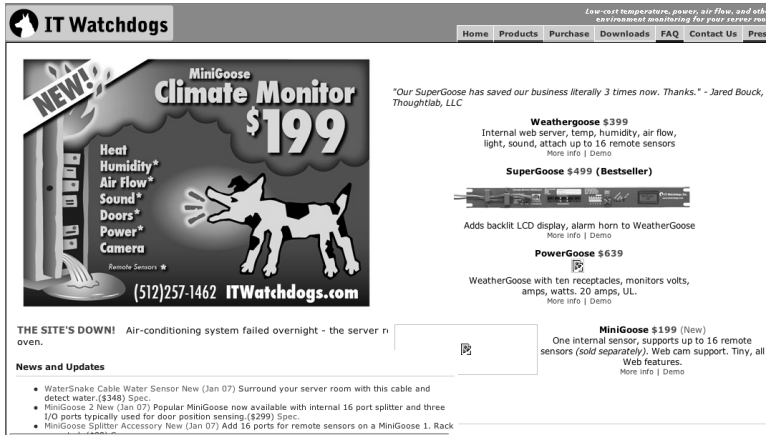


Figure 6

But you're anticipating the punch-line—ITWatchDogs grew every month, made millions of dollars, stole business from competitors with billion-dollar market caps (and professional-looking websites), and had a successful exit.

Of course it's only fair to also point out some of the **many instructive counter-examples:**

- Hipmunk[861] is the same thing as Orbitz or Travelocity—the *only* difference is amazing design, not just because it looks good but because it's so usable. In the words of Joel Spoksly[862]—the design "affords usability." (P.S. Early Hipmunk team member Alexis Ohanian[863] is so cool and smart and rich and funny and successful and good-looking that really he doesn't deserve to be alive. (P.P.S. Hey flamers, for God's sake it's a *joke!* Don't you realize I'm just sore from losing the Pecha Kucha competition[864] to him?))
- I always use and recommend Amy Hoy's time zone tool[865] *only* because it's *just nice to use and look at*. (P.S. she also authors a terrific blog aimed at the solo entrepreneur.[866])

- Many people credit Mint's smash success with their terrific design. Considering how many features were broken for how long, it's hard to argue.
- 37signals documented—with data[867]—how design changes results directly in more credit-card-swiping customers. It doesn't get more "business value" than that.

So where does that leave us in the "matters / doesn't matter" question of design?

Editor's Note: In 2025, fourteen years after this article was written, the Consumerization of the Enterprise and Product-Led Growth motions has brought design to the fore in business software, and consumers have shown a willingness to pick products based on design, and even pay more for great design. It's no longer clear that one can still hold the position that "design doesn't matter."

I think you can go either way, but you must **decide whether or not you're going to value design as core to your startup's identity, and then act consistently.**

It's clear from the outset that design is the *only competitive advantage* Hipmunk has over its competition. Specifically, by making the flight-search problem pleasurable and usable instead of feeling like you're navigating pivot tables from Excel '98. They don't have better data, better branding, better name, better SEO, or more money. Just better design, and not just easily-copyable incremental improvement, but a quantum leap better.

When design is fundamental to the strategy—how you *win* (p. 489), how you acquire and retain customers, garner attention and referrals, and be distinguished in the market—then design can be the most important thing (p. 1271).

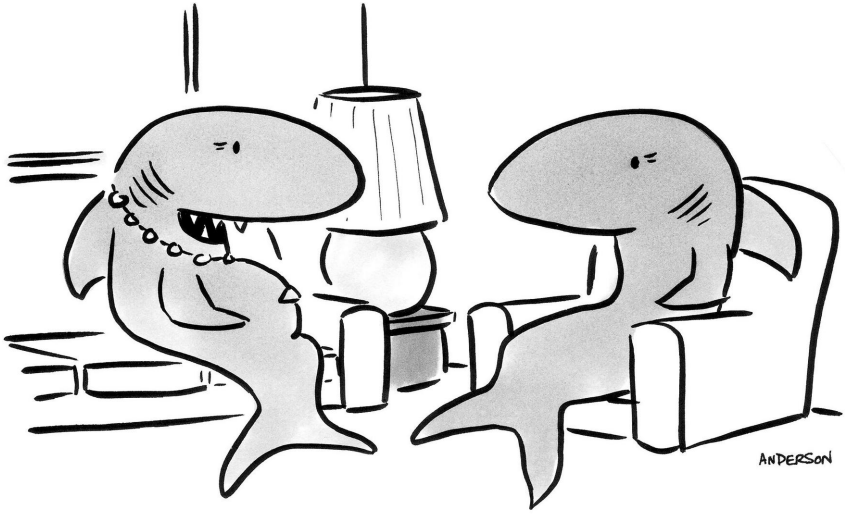
Conversely, at ITWatchDogs the company's internal and external culture was that "we're low-cost, friendly, approachable, regular people, who understand exactly what you worry about, exactly what your budget is, and we nail it." The site might have *looked bad*, but our message couldn't have been clearer.

All our competitors had these slick, corporate-looking sites, whereas the indifferent chaos of our website was not unlike our customers' offices which themselves were littered with assorted hardware corpses and solder stains. (And I don't buy the typical retort that "that's design too," because no designer on Earth would have created that website because it looks "relatable.")

Neither is it an absolute either-or whether design is important; WP Engine is a good example of this. We did fine before the redesign and we're doing fine after it; I'm glad we did it, if not for objective data, then for the subjective feelings.

But it is useful to **decide where you come down on the question of design** in your startup, because if it's important you'd better work on that right now and **develop a consistent culture** of valuing design through-and-through. Or if design is not important to you, you'd better decide what *is* important and **nail those things all the harder**, because you'll be competing with people who are using superior design to cover up their lack of competency in those areas. Your superiority must shine through regardless.

Chapter 62:
The only way to guarantee startup success



"As long as it has all its fins and is a remorseless eating machine, we'll be happy."

credit 868

Everyone said this would be the most embarrassing moment in the band's eight-year career.

Depeche Mode had decided to play the Pasadena Rose Bowl—capacity 60,000—for the 101st show of their 1988 tour. To sell out would make it one of the largest music concerts ever played in America—highly unlikely for an English electronic band. Claiming they were popular enough to fill that stadium was an audacious act of bravado that critics were eager to see transform to humiliation as they played to a vast, near-vacant space, mocked by the each of the tens of thousands of empty seats.

KROQ DJ Richard Blade knew that secretly the band themselves were doubtful. As he sat with singer/writer Martin Gore in the empty stadium before tickets went on sale, Martin "confided in me that he was nervous and hoped they could at least sell out the floor seats—just 10,000 tickets."

But the show sold out, with paid attendance greater than any Rose Bowl event in the eight preceding years. It was the defining moment of the band's career—the moment when they undeniably "made it."

And yet, simultaneously, one of the saddest moments.

Again from Blade:

> "Backstage, after their amazing performance, I chatted with [lead singer] Dave Gahan as he cried from pure happiness. He told me that the tears were because he didn't know if the group could ever pull off anything this great again and for him it was the most emotional concert of his career."

Indeed, they never would pull off anything that great again, even though they increased record sales, wrote more hits, released more albums, played more tours, even reforming the band after the loss of one member and through multi-year battles with drugs, alcoholism and depression from the three remaining members.

It's true though, what could top that moment? After you've proved everything that could be proved, to the critics, to your fans, and even to yourself?

What does it mean for you or me, that reaching the pinnacle of success is not only strikingly fleeting, but also unhappy? That tears of joy transmute immediately to tears of sadness, because **reaching the peak means by definition your next steps must be downhill**?

The same thing happens at startups. It's well-documented[869] that immediately following the "success" of an exit, founders almost inevitably fall into a sadness and even depression, as I've written about before (p. 1005).

It could mean this is all for nothing. That our fleeting moment of noon-time glory is cruelly bookended by years of gut-twisting emotional mountain-climbing (p. 737) in the a.m. and a meandering, permanently unfinished quest for meaning and purpose (p. 399) in the p.m.

It does mean this, if you let it. **The way you let this happen is to believe that the goal is to achieve a single moment of success**—a big sale, going public, or passing it on to your daughter.

Rather, you must understand that it is **the building**, not the result of that building, that matters.

Let's break it down:

You spend 99.99% of your time on the journey of building a startup, and 0.01% basking in the temporary euphoria of "success," such as selling it to someone else or ringing the bell on the morning of your IPO.

If that moment even comes, which it most likely will not. Often 100% of your time is spent on the journey, 0% in euphoric ecstasy.

If you look back and say "That was a wonderful time in my life. I'm glad I did it, and I'd do it again. Sure I would do some things differently—is that not true of anything?—but I'm proud of what I did and I'm stronger and wiser today than ever before. I created terrific jobs, where great people were empowered to build important things

together, who were comfortable and safe, who shared in the bounty of whatever upside we could muster, and who themselves would also say they're personally fulfilled just as I am."

Then you win.

But if your measure of "success" is based on a specific outcome—based on metrics, or money, or growth-rate, or number of employees, or whether you had an exit, or how much money you raised, or how many humans you cajoled into sticking their nose in your app, then mathematically you're almost guaranteed to fail, but only because of your own definition of failure (p. 1261). You're guaranteed to spend years of your life in nervous agony, chasing an outcome you think will make you happy instead of *making all those years **be** the years you are happy.*

Then you lose.

Don't get me wrong—I love making money, I've made plenty from previous startups and I hope to make plenty more at WP Engine, and so do our employees (all of whom are shareholders) and our investors. And honoring metrics is part of building a high performing team and building a huge, sustainable enterprise. Nothing wrong with using metrics and money to keep score in this game.

Keep score, so long as you can distinguish between the game and life. Keep score, while also basking in the thrill of generating happy customers and launching unique (p. 891) products and gathering the energy and brainpower of brilliant humans tackling interesting problems.

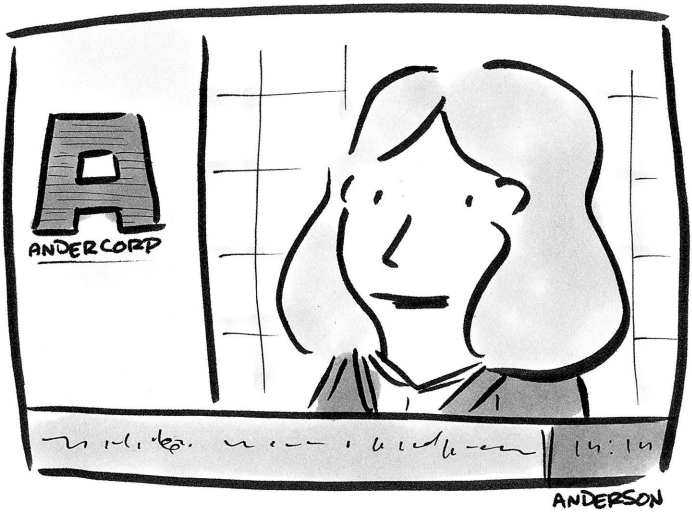
In fact, I'd argue that focusing on the nature of the journey (p. 827) solves one of the great riddles facing all startups. To paraphrase Peter Thiel: the first employee joins because it's the ground floor of an exciting startup, but why will the 20th person join? Or the 200th?

To be part of the journey.

That's why you should be there too.
If you value the journey, your "success" is guaranteed.

Chapter 63:
Deciding whether an investment is worthwhile

EV IS WRONG · OUTSIZED PAYOFF
ACCUMULATE VALUE · STABLE RETURNS · TRENDS
EDGE · VISION · RARE



"AnderCorp surprised investors expecting the worst today by performing completely miserably."

credit[870]

Whether it's your career path, which job to take for the next four years, investing in your personal brand, creating a startup, or investing in a startup, the question is:

How do I decide which investment is worthwhile?

I'm defining an "investment" as distinct from other bets by having these characteristics:

1. Spending time and money and reputation for months or years, hoping for a large return in the far-future.
2. Can do only 1-2 of these simultaneously (p. 901), because you are "all in."
3. Because the future is uncertain (p. 193), there is significant risk: Whether it will pay off at all, and if so, by how much.

This is in contrast to short-lived experiments, like a bet in poker, an A/B test, or making a brochure for a product concept to see whether people will buy it. Bets, experiments, and trials are excellent techniques, but by "investment" I mean a big commitment (time, money, reputation, opportunity cost), over a long time-frame, with a large potential payoff, under conditions of uncertainty.

Tackling this combination of constraints leads to a framework for evaluating which investments are worthwhile.

DON'T USE "EXPECTED VALUE"

Economics instructs us to use "expected value" to evaluate investments. This is almost always the wrong tool for the job.

The theory appears sound: "Expected value" means multiplying the value of the outcome by the probability of attaining that outcome, resulting in a single number that encompasses both risk and value; a "risk-adjusted value," as they say in finance.

Let's take an example to see why this number isn't useful for evaluating investments.

Investment "A" requires you to spend $50,000, and takes an entire year of your life. The payoff is $1,000,000. Furthermore, the probability of payoff is 100%—it's a signed contract with cash in an escrow account. Therefore, the "expected value" is **$1M**,* and the return on our investment is **20x**.**

Great investment! Now consider investment "B"—you also spend $50,000 and a year of your life, and this time the payoff is a whopping $10M, but the probability of that payoff is only 10%. The other 90% of the time, you receive no money at all. The "expected value" is also **$1M**,*** the same as investment A. And therefore investment B is also a **20x** payoff.

Perhaps you already see the flaw in the logic. With investment B we will almost certainly lose money and lose a year of our life, yet we're still claiming it represents a "20x payoff." We're also claiming that B is economically indistinguishable from A.

But unless you already have many millions of dollars in the bank, you probably wouldn't pick investment B, because investment A is a guaranteed life-changing amount of money (p. 45), while B is almost certainly a waste of a year and $50k. They are not even remotely equivalent.

It's even worse in real life because you rarely know either number—neither the risk nor the payoff. Whether it's the stock market, a startup, or whether a certain career path will still be lucrative in ten years, even experts can't predict (p. 193) probability or outcome. So, what

---

* Because `100% × $1M = $1M`

** Because the $1M expected value is 20 times the cost of $50k.

*** Because `(10% × $10M) + (90% × $0) = $1M`

garbage numbers are you multiplying together to get "expected value," and isn't the result even worse garbage?

There is one case where expected value is the right tool: A portfolio of bets. When a VC invests in twenty companies, when you write 100 newsletters and 1000 tweets, when you play 1000 hands of poker, when you have a balanced, uncorrelated financial portfolio, then you'll have some winners and some losers, and they roughly balance out, and therefore an "average" like expected value is indeed what you should expect. This relative stability in outcome is why a portfolio of asymmetric bets is a good strategy (p. 489). The types of investments covered in this article aren't like that—e.g. what job to take next, what startup you're going to spend your savings on—but if you can place a number of uncorrelated* bets, then expected value becomes useful.

MANDATE AN OUTSIZED PAYOFF

In the success case, the investment must have an enormous payoff.

If the payoff isn't huge, then it's not worth the (unknown!) risk. Not just the risk that it pays off at all, but the magnitude of the payoff. When it can pay off 10x or 100x, or when it can be life-altering, then even if you fall short of the goal—only 3x or only life-improving—then it was still worthwhile.

---

* Correlation is a key factor which often brings down portfolios, including angel, VC, and financial. "Correlation" means "as goes one, so very likely goes the other." In that case, you're back to the original scenario. Only if they are independent—like successive coin flips—can you employ "the average" to determine the likely total result.

> *"If riskier investments could be counted on to produce higher returns, they wouldn't be riskier."*
>
> *—Howard Marks*

How do you define "enormous?" How can you calculate the payoff, especially with things like career paths?

Use Fermi estimation. As I also suggest for ROI calculations (p. 171), probabilities (p. 997), and evaluating markets (p. 71), estimate to the nearest power of ten. Estimation in detail is difficult and often incorrect; estimating to the nearest power of ten is easy, often requiring little research.

For example, if one career path means earning 50% more per year for ten years (after you get established), that's 10x more earning power over your career. Or a startup targeting a small, stagnant niche might someday be worth $1M, whereas a small software company in a larger, growing space could be worth $10M.

You'll be tempted to round up; in fact I just did in the "50% more earning power" example. Don't do that, because you're exaggerating how good the investment is; typically if there's any error, it's in the direction of being less valuable. If, for example, your main career goal is to maximize your earning power, then "50% more" is not good enough; you need to find a different route that potentially doubles your current trajectory.*

This is the same logic as selecting Rocks in the Rocks, Pebbles, Sand prioritization framework (p. 221). Your large personal investments are Rocks, and must be evaluated accordingly.

---

* 2x more for 10 years is actually 20x of course, but now rounding down to 10x makes sense, especially since it might not be 2x exactly and it might take a few years to ramp up.

ACCUMULATE VALUE OVER TIME

For the payoff to be large, value needs to accumulate, which means it needs to be possible to add value reliably, and for value-already-added to not erode quickly. At a minimum, value must accrete, but ideally you construct a positive feedback loop, where growth or size itself begets more growth.

Social media and SEO are like this—the more you write, the more others interact with you, the more others link to you, the more attention you get, the more traffic you get, which makes it easier to get more attention and interactions. Writing consistently increases your ability to write well and often, and thus there is a positive feedback loop of growth and value.

Product companies are like this, especially with the recurring-revenue business model, where the revenue grows so long as each month there are more customers starting to use the product then there are stopping, which will happen so long as you're making a product that people truly want and love (p. 275). (Here's my roadmap to Product/Market Fit (p. 9) for how to build such a company.)

Consulting and professional services companies are often *not* like this. You'll have regular customers, but the reason consulting companies and accounting and law firms are often valued[871] around 1x revenue or less, while SaaS companies with similar growth-rate are valued at 5x–10x revenue, is because the work you do for a client isn't accumulating and applying automatically to all other clients, and consulting companies are hard to scale (p. 701).

> *"The most powerful force in the universe is compound interest."*
>
> *–Albert Einstein.*

Some people insist that investments must be *exponential*, but often they don't understand what that means, believing for example that "viral products" are exponential, when in fact they are not (p. 115). Exponential is wonderful of course, but not required, and in fact is probably a red herring for most ventures.

STABLE SUCCESS CONDITIONS OVER TEN YEARS

Just because the world values something today, doesn't mean it will be valued ten years from now. If you spend 3-5 years investing, expecting a payoff in the subsequent 5 years, the world had better still value that thing during years 5-10.

This summarizes Jeff Bezos's famous explanation of why Amazon's strategy has always been based on "what will not change." He articulates it so well, I'll leave it here without further commentary (except my emphasis, in bold):

> I very frequently get the question: "What's going to change in the next 10 years?" And that is a very interesting question; it's a very common one. I almost never get the question: "What's *not* going to change in the next 10 years?" And I submit to you that that second question is actually the more important of the two—because **you can build a business strategy around the things that are stable in time**. … [I]n our retail business, we know that customers want low prices, and I know that's going to be true ten years from now. They want fast delivery; they want vast selection.
>
> It's impossible to imagine a future ten years from now where a customer comes up and says, "Jeff, I love Amazon; I just wish

the prices were a little higher." "I love Amazon; I just wish you'd deliver a little more slowly." Impossible.

And so the effort we put into those things, spinning those things up, we know the energy we put into it today will still be paying off dividends for our customers 10 years from now. **When you have something that you know is true, even over the long term, you can afford to put a lot of energy into it.**

—Jeff Bezos,[872] 2012 (it's been 10 years since this quote, so we can confirm in retrospect that this was indeed the correct analysis.)

"Second quarter is up. No, fourth quarter. Now second again. OK, it's third now..."

credit[873]

ALIGN WITH GLOBAL TRENDS

If the best investment is in something that doesn't change, the next best is aligned with a global trend with sufficient momentum that it is unlikely to change soon, if ever.

For a startup, you want to be in a large, growing market. This maximizes the number of niches you might find traction in, without triggering a reaction from a large incumbent. It maximizes the number of marketing channels you might be able to leverage. It means people are already spending money, so you're answering "why buy from me" rather than "why should I allocate budget for this at all" or arguing against "that's a problem but not a priority" or evangelizing against "I never heard of that problem." It means the pie is growing, even if your percentage of the pie is small.

Other current trends are the increase in virtual and distributed workforces, the increasing average age of humans on Earth, the increase in physical delivery services and online shopping, the increasing global demand for health care, education, and energy, and of course the rise of AI in unlocking novel features and capabilities for products and services that are changing the nature of work and life.

> *"The stock market is a device for transferring money from the impatient to the patient."*
>
> *–Warren Buffett*

A trend you can depend on is either (a) so large today that it would take a decade to change significantly, or (b) is growing so fast (e.g. at

least 5x in next 5 years) that it will be a strong trend even if your growth estimate is significantly wrong.

HAVE AN "EDGE"

Why should *you* make this investment, not someone else? A good investment, that *you* cannot execute with excellence, isn't a good investment for *you*.

What's your edge?

In financial or startup investing, the edge is often your ability to evaluate the risk. You might understand that a certain execution in a certain product in a certain market carries lower risk than others know. This is an edge, though it is a rare one, in part because even experts are bad at evaluating risk (and the majority of VC funds lose money,[874] and Angel investors are worse).

More likely it's something special about your execution. I'm avoiding the word "unique," because although it's great if you're truly unique in some way, this is rare, and in fact unnecessary. You should, however, leverage your strengths (p. 543) so that you have an edge, despite also being worse (p. 891) in most other areas.

One way to be special is to be the top 0.1% in the world at something (e.g. Olympic athlete), but that's probably not what you are. The other way, and the way available to all of us, is to be in the top 25% at a few uncorrelated* things, which together creates a special edge.

The quintessential exponent of this idea is Scott Adams, writer of the Dilbert cartoon and now political activist, who points out[875] that he is a decent illustrator but not great, a decent humorist but worse

* For example, being great at both math and computer programming doesn't make you any more special, because those often occur together, whereas being artistic and great at programming places you in a much more elite group.

than any comedian on Netflix, and has held jobs in the tech sector, which almost no artist or comedian has done, and therefore it is at the intersection of those three "circles of competence" such as that he gives something unique to the world.

"Passion" can be part of an edge, but it is not enough. Passion creates motivation, which is especially important when you're persevering (p. 429) through difficult times. However, having passion doesn't mean the risk is low, the payoff is high, the market is good, the customers are there, the product is desirable, or the career is lucrative.

Having an edge reduces risk while increasing not only the potential size of the payoff, but your own personal fulfillment (p. 399), because it puts you at the center of what you love, what you're good at, and what the investment needs to be successful. If you're not sure what your edge is, here are some questions to ask yourself (p. 569).

ALIGN WITH YOUR LONG-TERM VISION

If you run quickly in the wrong direction, that's not progress.

If you make investments that pay off, but the pay-off isn't what will make you proud (p. 827), or happy (p. 1005), or fulfilled (p. 399), or rich (p. 45), or whatever your goals are, then it's not a good investment.

If you don't have a long-term vision for yourself, consider making one. Taylor Pearson[876] has some great advice on how to make a vision[877] and how to set short-term goals[878] against it. See also the last section of my article on fulfillment (p. 399), with a series of questions that help you discover what you love and hate.

In much of my life, I haven't had a vision. Fortunately, my path has largely been in the right direction anyway, but I wonder how much of that is due to luck (p. 449).

Best not to leave it to luck.

GOOD INVESTMENTS ARE RARE

With all these restrictions and rules, it's hard to find a good investment!

> *"Given our size, we see few good things [to invest in]. If we were smaller, then we'd see lots of good things."*
>
> *–Warren Buffett*

Yup. Most startups fail. Most VC portfolios lose money.[879] 80% of day-traders lose money.[880]

That's why it's so important to adhere to these guidelines—because most investments aren't right for you, and you don't want to spend the next five years doing the wrong thing.

Of course you'll never know (p. 159) if you made the best choice, even with hindsight (p. 1253). So the goal isn't to make the "best choice," but rather to make an excellent choice. To invest wisely, with the information you have.

Good luck!

Chapter 64:
“Stealth mode” and other f’ing brilliant strategies

STEALTH · PERFECTION · DON’T ASK · RELAX
CODE · RAISE · COPY · EVERYONE

“Bad news – something that happened in Vegas has been spotted in Des Moines!”

Here are some of my super-secret winning strategies from 25 years of building startups.

STAY IN STEALTH MODE UNTIL THE LAST MINUTE

The last thing any startup needs is people finding out about it.

You can get attention later—that's easy. You don't need the distraction of all those customers clamoring outside your office, begging to give you money, while you're still refactoring your NoSQL database schema.

It's instant death if a competitor were to launch at the same time as you, so you're right to keep all your ideas completely secret. Once you launch, *then* millions of people will know about you, including competitors, but by then you'll be a full 4 months ahead of the whole world, making competition impossible, even if they are two Stanford kids with $9,000,000 in funding and the wind of YC at their backs.

That means you can't talk to potential customers (p. 239) either, because the word might get out! Worse: some of those potential customers—the ones willing barely willing to part with $20/mo—might rather save that money by quitting their day jobs and start a brand new company to rip you off. Their feedback isn't worth the risk!

Oh and *absolutely* don't talk to other entrepreneurs or advisors who can and surely will copy your idea. Just read the history of any startup or interviews of famous founders[882] and you'll find one thing in common—no one had mentors or found value vetting ideas and brainstorming with people who have trod this path before.

GET IT RIGHT THE FIRST TIME

With those millions of customers anxiously awaiting your launch like the next book in Games of Thrones, you absolutely *must not* disappoint them with a shaky v1.0 (p. 101).

Look around—other software companies wait until there's no bugs before they release. That's why they have no usability problems nor lack important features. With all those fully-completed products, it's a mystery why "feedback gathering" tools or bug trackers even exist. I guess some people just enjoy project management. There's no accounting for taste!

An ounce of prevention is worth a pound of cure. That makes prevention, like, 16x more valuable than cure. So take your time before you release, even if that means years. Perfection is worth the wait.

DON'T ASK ANYONE IF THEY'D PAY FOR YOUR PRODUCT

Of course they'll pay; the math is simple:

> They'll save 45 minutes a day by using your product. Even if they value their time at only $20/hr, and work only 20 days per month, that's a savings of $300/mo. Your tool is $20/mo, so this *is a money-printing machine for customers!*

Customers can't afford not to buy it. Maybe they should buy it twice.

That's how buying decisions are made—cold, rational, and based on micro-economics 101—so why waste your time and theirs, just to verify the obvious?

DON'T WORK TOO HARD

Building a startup is hard enough—don't make it worse by working too much (p. 1547).

All the great startup founders are known for 30-hour workweeks. It's one thing to be passionate—that's great of course—but that doesn't mean you should be waking up at 2am in a cold sweat. You need your sleep!

Steve Jobs didn't work constantly, Bill Gates had lots of hobbies, Mark Zuckerberg wasn't tethered to his laptop, and Tim Ferris really did become a best-seller by writing and then promoting his book while working just 4 hours each week.

Startups don't require obsession—that's an unhealthy rumor perpetrated by all 2,000 startup founders ever interviewed on Mixergy.[883] They're all lying—they actually lead healthy, balanced lives. They don't want you to know their secret, because this keeps potential competition at bay.

Don't fall for it. A startup is a job just like any other—you can leave work at work and make sure to use up all your vacation days!

WRITE THE CODE FIRST

Writing software is tricky. You've been doing it for 8 years, so this is the part of the business you know best—and you know the difficulties that await you!

The easy part of this startup will be getting attention and making sales. Getting people to a website is easy—it's not like there's 1,317,681,984 other websites clamoring for attention. Getting them to buy once they hit the website is even easier—why did they come to the website if they didn't want to buy? Getting consistent attention from the media is easy too—why *wouldn't* popular influencers want to talk about you all the time? Getting reseller deals is simple[884]—why wouldn't they want to make more money, in the grand tradition of the win-win (p. 627)?

No need to work on that end of the business, because a great product sells itself. The world isn't littered with failed startups (p. 379) with decent products but only four customers.

Rather, you need to focus on coding (p. 635)—the one part of the business you understand and have the most confidence in. Double-down on what you know!

Don't face your fears now. If you shut your eyes and learn a few extra keyboard shortcuts in Visual Studio, everything else will work itself out by the time you get there.

RAISE MONEY BEFORE YOU HAVE CUSTOMERS

Every successful startup raised a big pre-pre-pre-pre-seed round before they had customers or revenue. They had to—how else can anything get done?

Who cares if you already gave away control? You need ten engineers right now to build dark mode.

It's easy to raise money anyway—just take your a Figma mockup and $48B TAM justification to a few pitch competitions and three cocktail events on Sandhill Road. They won't ask questions about those meddlesome customers or revenue.

COPY THE COMPETITION

Why reinvent the wheel when successful competitors have already figured everything out? The smart move is to just copy their feature list. Or just 70% of it—whichever 70% seems easier. Customers probably don't care about the rest of those silly things, especially if you're 20% cheaper.

Speaking of pricing, copy their pricing. Because the competition is always smart and researched with optimal pricing, and is targeting the same segments in the same way with the same brand, so you can copy their pricing instead of talking to customers and figuring it out for yourself.

Speaking of talking to customers, don't waste time talking to customers! The competitors obviously did all that research already. Their product decisions must be right—that's why they have market share!

And forget about differentiation. Just match their features but 20% better and 20% cheaper. Customers will flock to you immediately since you're basically the same but slightly improved. Whatever "better" means.

SELL TO EVERYONE

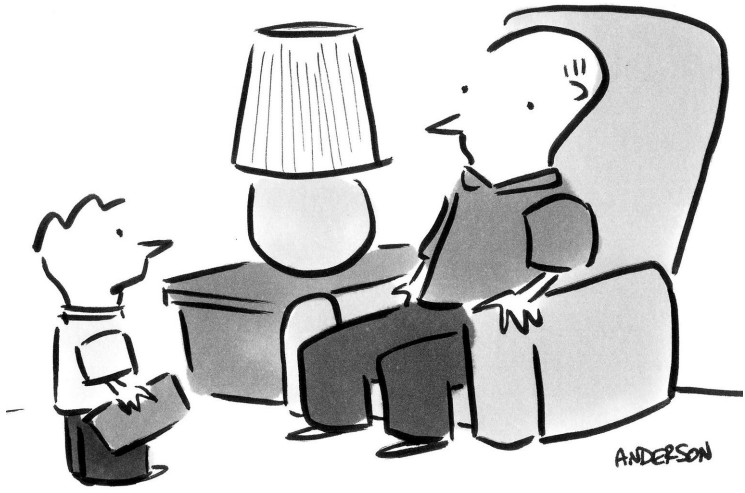
There's no reason why everyone on Earth wouldn't like your product—it's so great! Of course it is—otherwise you wouldn't have bothered to build it.

Besides, if you target one segment of the market (p. 317), all you're doing is alienating other people who would have bought it. If you don't say "everyone" on the home page, you'll lose sales.

Sure, other products make different trade-offs in features, usability, integrations, design, and price, but nothing is as right-headed as yours! That's why you made it all by yourself, without a team, without talking to anybody, so you wouldn't be distracted. No reason to wonder why people buy from all those competitors. You know the answer: they bought from competitors because they had no choice. Now that they can choose you, they definitely will.

Now that you have all my secrets, go out there and kill it!

Chapter 65:
Satisficing vs Maximizing



"Before you see my grades, I'd like to remind you that I lowered expectations for the fourth quarter some time ago."

credit 885

"Maximizing" means finding the best solution. It requires exploration and analysis to ensure "the best" option hasn't been overlooked, and that we have confidence in our evaluation of the options (p. 603).

"Satisficing" means picking the first or easiest or least-expensive option that satisfies the requirements. Preferring a faster decision to the best decision. It means not getting paralyzed by the pursuit of "perfect," but as a result, rarely results in the very best solution.

People naturally tend to be Maximizers or Satisficers, although it depends on the subject. For example, you might maximize your career, but satisfice your diet.

Studies show[886] that although Maximizers make better choices Satisficers *enjoy* their choices more, and of course spend less time and create less stress in making the choice.

You must choose when to Maximize, and when to Satisfice. It might seem like Maximizing is best, especially when you have teams of smart people who can do the maximizing. But often this is the wrong thing to do.

Speed is one of the greatest competitive advantages, partially explaining why small up-starts are able to beat large incumbents. Speed requires satisficing, not maximizing. Strength in startups comes from *shipping*, with benefits accruing today instead of theoretical benefits in future, using customer reaction and real data to decide your next move, rather than planning many moves ahead, which you can't do anyway (p. 193).

Satisficing is also best for "sand" in the "Rocks, Pebbles, Sand (p. 221)" analogy, when deliberation is the enemy.

Still, maximizing is best for some decisions, particularly decisions that should be made slowly (p. 745). Your Product & Business Strategy is a slow obsession, best when taken from many angles, pounding it with devil's advocacy, questioning every assumption, "sleeping on it" repeatedly, repeating pieces of it to customers, strangers, anyone who might have a useful reaction. Strategy isn't something you want to change often, with entire teams and careers being built on top of it, so maximization is best.

For the most important decisions, maximize with Binstack (p. 603). For almost anything else, satisficing is likely wiser. Be careful; the same perfectionism that makes you a force to be reckoned with (and that drives your impostor syndrome (p. 457)) will lead you to maximize too often.

For satisfying activities, define an objective threshold for "what is satisfactory." When a metric (p. 645) is below its threshold, investment is warranted. Otherwise, you *must not* continue to invest in it, because definitionally it's not valuable (enough) to deserve attention, whereas other areas can leverage that investment, either because they're below their own thresholds, or because they are things to maximize.

Pick the right goal for each decision, and communicate it clearly.

Chapter 66:
Worse, but unique

"I'm not much for video games, but I like the way he moves."

A perfect storm created a renaissance in chess: COVID driving people to occupy their minds in physical isolation, the hit Netflix show "The Queen's Gambit," and a surge of chess Twitch streamers and YouTubers fueling that nascent interest in chess with equal parts education and entertainment.

And so beginners learned openings like the titular Queen's Gambit —one of the most popular ways for white to open the game. It's easy to find courses, books, and online content about the Queen's Gambit because it's been popular for more than a hundred years. The top chess players in the world today still play the Queen's Gambit; if they play it, it must be good! And it is.

Except, it's smarter for beginners to play worse openings.

"Worse" is an objective evaluation. There are openings, like the *King's* Gambit for example, which were popular in the 1800s but have since been "refuted," meaning that for each move by white, black always has a counter-move which results in black ultimately achieving a superior position. "Superior" as consistently demonstrated in games between top-level humans, as well as computers that are orders of magnitude stronger than those humans. Thus, the King's Gambit is objectively worse than the Queen's Gambit.

So why should beginners use the refuted King's Gambit opening?

First of all, beginners aren't playing top chess players; they are playing someone near their own strength. At their level, the King's Gambit isn't refuted, because neither player has memorized the sprawling branching sequences of all possible moves and counter-moves and counter-counter-moves, *ad nauseam*. Nor can they figure out the best move in the moment. Anyway, the game will be decided by stupid blunders (p. 379), not brilliant tactics.

Second, and more importantly, exactly because the Queen's Gambit is popular, it's likely that your opponent has also studied it. So it's quite possible that your opponent knows a defense to the Queen's Gambit better than you know the attack of the Queen's Gambit. As you progress in chess, this becomes more and more likely. So now the

question becomes: Is your preparation—memorization of the branching forest of moves—better than your opponent's preparation?

That's not the sort of "luck" you want to depend on. That's not a good strategy.

The solution is to learn a refuted-but-not-completely-atrocious opening like the King's Gambit. It isn't refuted at your level, and also there's almost no chance your opponent has memorized any specific defense for it, because "no one plays that." So they're reacting to it for the first time, surely not finding all the best counter-moves in the moment. But you *have* memorized a few of the best moves. So you have a clear advantage, even though you're a beginner.

This is why leverage, as defined here (p. 543), results in a superior strategy for products and companies, even if many things about you, your team, your product, and your company, are "worse" than the competition. Playing on your own terms changes the game. "Worse, but unique," is in fact "better."

A startup is a "beginner" in the market, facing a variety of "opponents" who have varying degrees of preparation and skill (age, capabilities, customers, brand, funding). A startup needs to fight the competitive battle on its own terms, not attack where the opponent is already strong.

Even on a personal level, you must use your leverage to do something special, be something special, at the intersection of what is special, enjoyable, and what the world needs (p. 399).

> *"Be yourself. Everyone else is taken."*
> *—Oscar Wilde*

Many potential customers won't want to play your game, and you'll feel bad when you lose them to a competitor, and when they say mean

things about you and your product and your decisions. Especially when it's in public, on review sites and social media. But don't worry, you weren't going to win them anyway—they're just not into you.

The customers who *do* want to play your game, will see that you're the best in that game. Even if you're two people and there's a dozen competitors ranging from well-funded, well-covered-by-the-media startups to old-but-reliable incumbents. If you pick a different game, you're the best at that game, and you can become extremely successful.*

Even better than "different," is to be extreme in that difference. Not just a minimal UI, but so minimal it works on the command-line. Not just great design, but so remarkable people buy it only for that, and it's written up on designer's blogs. Not just a new algorithm that solves an old problem, but one that uncovers new things that no one else does, even at the expense of missing things that others catch. You can't do this for all aspects of your product and business—indeed, even a single one is already powerful—but extremity is how you maximize the power of the few things that make you special.

All those competitors *cannot* play the game as well as you, because you picked something that was not only different, but extreme. They can't make every choice you made, and since they didn't, they're definitionally not the best at your game.

Play your King's Gambit. Don't worry if others claim your strategy is refuted. It's not refuted for you.

Indeed, the fact that they *think* it's refuted, is why you will win.

---

* In my case, WP Engine powers as many substantial websites as Google Cloud or Digital Ocean, and we still have only 2.3% of that market. That means 98% of potential customers are in fact not customers, and yet WP Engine is a unicorn. By the way, this is another reason why it's wise to seek large markets (p. 71).

Chapter 67:
Discount gambit

"No, it's fine, I've just never seen coupons used in an acquisition."

Which of these pricing strategies is more persuasive?

1. If you buy now, I'll give you a discount.
2. The price is going up, but if you buy now I will lock in your rate.

Both are types of discount. **The typical software sales strategy is #1.** It's often applied to get the customer to "close" before the end of the month or quarter or some other arbitrary time boundary.

At first blush it seems harder to persuade with #2. After all, #1 means the customer pays less than #2, because #2 isn't a real discount —it's a discount against an imagined future price.

But for me, **the evidence is overwhelmingly in favor of #2.** Here's why, from the point of view of the customer.

In strategy #1 there's a discount if I "act now." Hmm, so that means the "price" wasn't really the price after all. The "price" must have included a nice slice of pure profit that apparently you're willing to forego. So you were gouging me before? And the only reason I found out about it is that it happens to be the end of the quarter?

Thus **#1 breeds mistrust**—the opposite of what you're trying to establish with me, your customer. **In #2 you're looking out for my interests.*** You're cluing me in that there might be a rate increase, and you're actively protecting me from it. Sure, I know there may not be an increase, or it may not come for a while. But it's still protection, not a gouge that you graciously chose to reveal.

Four years ago I was trialing software called `.TEST` from Parasoft. It was buggy; even after hours of remote desktop control with tech support we couldn't get it to stay up long enough to scan my code.

But the salesman was persistent. The conversation went like this, minus many minutes of sales-filler-language on his end of the phone:

> "How much will this cost me?"
> "$20,000."

* This is an example of the "Love" type of WTP (Willingness-to-pay) (p. 275)

> "Wow, I thought you were going to say $2,000. That's way out of my price range for one person and this product. In fact, I've looked at FxCop and NUnit and it seems to me I can do the same thing with free tools. I was willing to pay for some convenience, but not that much."
> "Let me see what I can do."
> "No nevermind, it's, like, an order of magnitude problem."

He called back the next day.

> "$1,500."

I didn't buy. I talked to someone who did, though. A reference customer. That guy said he paid $20,000. I asked how he liked it and whether he encountered the crash problems I was seeing. He said they hadn't installed it yet, but the demo looked great. I made a mental note to try to understand the mentality and budget that forks out $20,000 for a nice demo.

But getting back to the point. If he can go from $20,000 to $1,500, maybe he will go to $1,000. Yes, this strategy means often you will extract extra money from me. But it also means I don't know where the floor is, and I have every incentive to haggle. The process drags out, ending at gunpoint. **Meanwhile your "customer relationship" is now more of a "hostage situation."**

So let me get this straight: It's better to get an extra 10% on every order, but create an adversarial environment with me, your cherished customer? This is enterprise sales, right, where the pilot is 30 seats and the roll-out is 2,000? And you're going to risk pissing me off over 10% on the 30-seat part?

And now imagine if I had called back that reference customer and told him he could've had it for $1,500? Yet another problem with discounting—word gets around, meaningless differences in pricing is unfair, and now I, the customer, see you as plain old dishonest.

Goodbye 2,000-seat order.

Even if we set the honesty/relationship argument aside, there's the matter of image.

What kind of company provides a discount? Wal-Mart, Target, Walgreens. Try to get quotes for Microsoft or Oracle or IBM products for 1000 desktops. Everything's negotiable, everything's discountable. At best it conjures images of haggling and struggle; at worst of low-quality or the desperate need to "meet numbers" at the expense of everything else.

Which companies don't discount, ever? Apple, Google, Constant Contact. No discounts on iPhones. No haggling over AdWord prices. What's the image? Desirable. The best. Worth paying for. **The leader doesn't have to compromise. The leader isn't desperate for orders.**

What kinds of customers do you attract through discounts, coupons, and "act now" language? The ones that truly find a lot of value from your product? The ones whose opinions you care about when you're researching the next round of features? The ones who will love your brand, and increase how much they're spending with you over time? Strategy #1 attracts the least-valuable customers and repels the best ones, while establishing a weak brand.

Strategy #2 implies growth. You've planted "higher prices" in my head now. "Supply" in software is unlimited, so that must mean "demand" is increasing. The customer won't go through that calculus, but it certainly feels like the product is becoming more valuable. Discounts feel like unloading unwanted product; price increases feel like success.

Strategy #2 implies I'm part of a club. I've gotten in early, on the ground floor, before the product explodes in popularity and prices go up. And I'm rewarded for this support and loyalty with price protection. A "thank-you" from because I was part of it, because I was there before you were big and expensive, because I took that risk with you. That's actually a good reason to give some a discount, and here you are, proactively giving it.

So there it is. #1 means less money now, an adversarial relationship, a never-ending struggle over money, and a message that maybe

the product *needs* a discount to be desirable. #2 means more money now, a consistent and fair pricing policy, an inclusive, special customer relationship, and a message of market leadership and growth.

So why do 90% of software companies pick #1?

Chapter 68:
You can have two Big Things, but not three

"Unfortunately, not only have they stolen your identity, they're also living your best life."

Forget work/life integration for a minute. How much time do you have, regardless of partitioning?

From your 24-hour daily allotment, the 1950s-style break-down is 8 hours for work, 8 for home and commute, and 8 for sleep and ablutions. So, "work" and "home" are the two things in which you can spend 40+ hours per week.

This is the amount of time it takes to tackle something huge. A career. A parent. A startup.

There are weekends and vacations and sick days and such, but those don't add up to enough *concentrated* time to carry off something like a startup without causing work or home to suffer.

Of course "work" and "home" are just placeholders for "Big Things." If you're unattached, "home" doesn't occupy significant time.

The rule of life is: **You can have two "Big Things" in your life, but not three**.

Big Things include:

- Job
- Kids
- Spouse
- Social Life
- Major Hobby (e.g. build a boat in the garage, become a chess master, video game addiction)
- Startup

You can do a startup on the side while you have a day job, but your family will never see you. You might even lose your family. It happens. This is partly why it's easier to start a company before you have a family or even a spouse.

You can have a job and a social life, but unless your spouse is fully integrated and agreeable to that social life, there will be strife. "Going out with the guys *again?*"

"Someday we're going to look back on false memories of this and laugh!"

credit 890

Yes, "kids" and "spouse" are on the list separately. Young kids strain marriages because there's not enough time to invest in the kids as well as be there for each other.

Some people try to "have it all." Men and women both. But it's never true. At most two can function well; the rest do not. More often, there's just one that receives the majority of the energy, and the rest suffers.

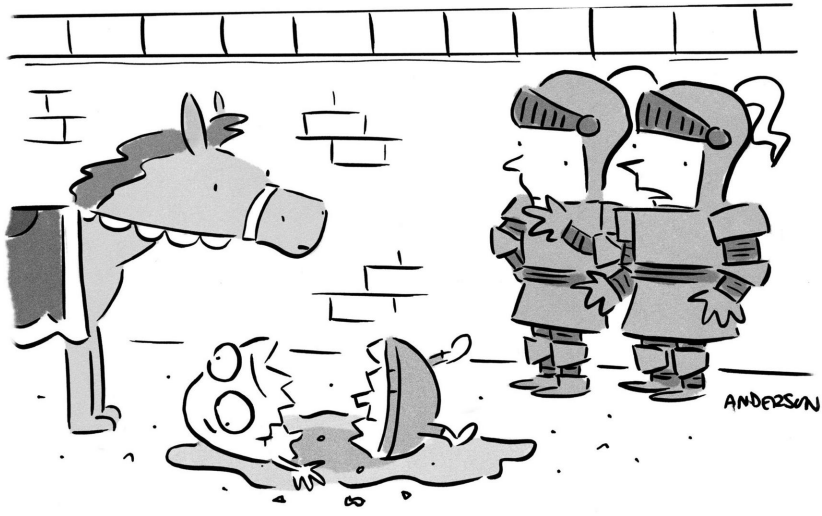
Note that "Sleep" isn't on the list of options, even though it's mathematically the same in terms of time commitment. That's because cutting out sleep doesn't work—then you can't function at a high level at *anything*.

No, you are not an exception. That's egotistical self-deception. Not on sleep, and not on the number of Big Things. Ask the people around you if *they* think you're not failing at one of your Big Things.

Time to decide which two.

Chapter 69:
Brittle Points: How to make companies robust

BRITTLE SERVER · HARDENING · BRITTLE SELF



"I'm less interested in putting him back together, and more concerned about the giant chicken who put him up there."

credit[891]

Companies fail for all sorts of reasons (p. 379). Sometimes we're not surprised, like if the product never worked or not enough people wanted to pay to solve the problem in question.

But sometimes the product did work, people were paying to solve the problem, sales were starting to pick up, and then something broke, and it was all over. How does that happen?

- The product was built on a platform,[892] and the platform changed. A popular app drops to zero downloads after Apple builds it into iOS; a Google Workspace add-on drops to zero sales after Google builds that feature into Docs; a Twitter management tool breaks when Twitter removes functionality from their public API.
- The initial marketing channel quickly saturated (p. 1369), so growth stalled (p. 1191) at a non-zero but unsustainably-low rate.
- The initial marketing channel was sustainable for a while,[893] but got wiped by external forces: large bidders tripled the cost per click (p. 1377), Google's SEO algorithm changed, the big industry event stopped happening, the link-sharing site became irrelevant, the hot blog lost its traffic, the magazine running the ads finally closed.
- One big customer,[894] representing 80% of total revenue, left. It wasn't a mistake to sign that customer—it funded the company.
- A key employee left the company. Early on, a 10x person[895] can mint the company but also could be irreplaceable. A suitable replacement is too rare; it takes too long to find someone, convince them to join (p. 1417) for almost no salary, and get them up-to-speed and productive.

I call these "brittle points"—places where sudden changes mean the company catastrophically fails, regardless how wonderful everything else is.

All young companies—and some mature ones—have dependencies like these. You can't help it; you have to rely on other technology to

build the product, services from vendors to deliver the product, and human beings to do work.

Engineers have Brittle Points in their infrastructure, so they've developed common patterns to address it. Let's briefly look in how they do that, because it will give us clues about how to solve the problems in that list above.

A BRITTLE SERVER

Suppose we have a single server that runs our website. Any number of things can cause this server to break—a power failure, a network failure, a bad configuration change, too much traffic arriving at once, bugs in the code, all sorts of things.

How do we make this system less brittle?

Consider power failure. Power can fail if the power supply* inside the server burns up, or the power strip fails, or the power cord fails (maybe through a wetware[896]) failure like accidentally unplugging it).

We can address this Brittle Point with a second copy of the power components—a second power strip with a second cord plugged into a second power supply. This is, in fact, exactly what data centers do!

In short: redundancy—having two things that do the same job. **It's twice as expensive, but it buys robustness.** This is also what airplanes do (p. 1267).

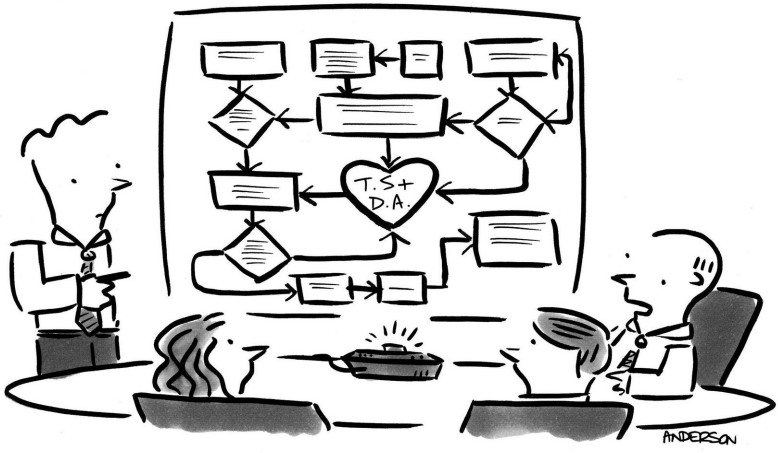
But what happens if the city power fails? Data centers have their own gas-powered generators. Which means they stockpile gasoline. Rarely-used gas-powered engines tend to fail, so they also test and maintain those units weekly. Data centers often have multiple generators. More robustness, purchased at significant, on-going expense.

* This is the component inside the computer that receives the power cord; it converts city-power into the type of power needed by the other components.

In modern clouds we go yet another step further, because the entire data center is itself another Brittle Point. So we have additional, identical servers in a physically-separate data center, that draws power and network connectivity from different outside vendors. But now we also need a smart networking system that knows how to direct internet traffic to only the servers that are currently available. Yet another system, which itself becomes—you guessed it—a Brittle Point.

The pattern continues—fix more Brittle Points, at more cost and complexity, sometimes creating new Brittle Points. Reliability is expensive.

This pattern is applicable to all of the causes of failure above.



credit[897]

"OK, but what happens to the supply chain if you and Debbie break up?"

NEUTRALIZING BRITTLE POINTS

"One platform" is brittle, because if the platform-owner forward-integrates (i.e. copies you), or removes APIs that you depend on, or themselves fail, that's the end of your company. One solution is to be multi-platform.* Another solution is to only build on platforms where you have a high degree of confidence that the platform owners are committed to supporting their ecosystem by never directly competing with them. Ideally, the platform even promotes their ecosystem, so that it becomes a growth vector instead of a Brittle Point. (Salesforce is currently the best in the world at this.)

"One marketing channel" is brittle, because if anything happens to the channel, that could be the end of an otherwise-healthy company. The solution is to find additional marketing channels (p. 115), so that variation in any one of them is not fatal. Of course this also creates growth; again this "double-win" of "growth + risk-mitigation" shows why it's especially valuable to invest in.

"One big customer" is brittle. One solution is a long-term contract with a serious breakup clause—insurance to bridge the time it will take to win replacement customers. Another solution is to prioritize sales until that customer represents a smaller percentage of revenue. Another is up-front payments (p. 353), so you have the cash-flow to invest in that growth. Another is to charge even more than you originally calculated, again for mitigation cash-flow. The typical attitude is, "We now have a large customer, so pour extra money into development to make sure we don't lose it," but the right attitude is to use that money to win other customers.

"One key employee" is brittle. Not only might they leave, but they will inevitably get sick or take vacation. The usual refrain in the start-

* Example: At WP Engine, we run on all three major clouds—AWS, Google, Azure. Example: A marketing tool for listening and responding on Twitter could add support for LinkedIn, Threads, and Bluesky. In this way, additional features can also be risk-mitigation.

up world is that none of these are options—everyone has to work 70+ hours/week (p. 1547) to the exclusion of other things (p. 901). Talk about brittle!

Solving these things takes time and money. Like the server example, they're not free, and not quick-fixes. You can't just hire three more fantastic developers to create a robust engineering team, and you can't just snap your fingers and find three new efficient, productive marketing channels.

Therefore, the right attitude is to **maintain a list of these risks** and then periodically ask (p. 1065): **Which single one is best to attack right now?**

For example, it's cheaper and easier to experiment with new marketing channels than it is to find, interview, convince, and manage a second software developer, and plus if you can get a second marketing channel online, that will generate revenue, which in turn means you can *afford* a second software developer. In this mini-scenario, the best thing is to focus all your energy on getting a second marketing channel working.

As you scale up (p. 773) **the size of the "chunks" that create brittleness also scale up**, which creates new brittle things, and thus new risks and new investments. For example, with $260M in revenue in 2016, still growing at a blistering 60%/year, with a thousand employees, Hubspot was not brittle in any of the ways outlined above. But they recognized that they were a single-product company. At that scale, that's a Brittle Point: if there were a sea-change in the market for inbound marketing software, that could be fatal to Hubspot. It also limits long-term growth (p. 1191) as the market matures and saturates. The way out—the redundancy—was to become a multi-product company. Furthermore, the second product had to scale at least as well as the first one; it's still a Brittle Point if one product is 95% of revenue. They attacked that problem, and in 2024 it's clear that they succeeded.*

* See this article (p. 335) for data and discussion.

THE BIGGEST BRITTLE POINT

Finally, on a personal note, there's another "chunk-level" that's even larger than all of the preceding, and **it's a brittleness that almost all founders suffer from, including myself**. The chunk of "the entire company."

This is a one reason why founders are almost always sad and sometimes permanently depressed after a successful sale of a company (p. 1005). This was your identity, your life, for years. You don't remember what it was like to "be you" without it, and anyway "you" aren't the same "you" anymore. You don't have hobbies or even good friends anymore. You might have sacrificed family or health (p. 1547). Talk about a Brittle Point. Your entire life is a Brittle Point.

The solution here is not to have two companies or two jobs. That's burnout; a lack of singular focus creates worse outcomes (p. 901).

Rather, the solution is to realize that there were things you did and loved before and there will be things you will do and love after. They might not all be the same things. Sometimes it's best if they're not; you're a different person now. You are inside a chapter in the book of your life. Even if one chapter is sad or has an unexpected twist, there's the next chapter which you can look forward to, even if you don't yet know how that story will unfold.

You can rediscover who you are with this process (p. 569). But it's rarely easy, or simple, or fast. Hopefully the successful sale has literally bought you time.

Robustness, not in many things simultaneously, but in things serially. That's what you do with limited time, and how you navigate the arc of your life's story.

Back to today and the here-and-now. Go list all the Brittle Things you have today. Then tackle one or two of those things at a time—you have to manage risk (p. 667), not try to eliminate all of it at once.

Be thoughtful, and build steadily away from brittleness.

Chapter 70:
p-Hacking your A/B tests

FAKING RESULTS · FAKING A/B TESTS · P-HACKERS
FOOLING YOURSELF · THEORIES

"Fine, just stand there being ominously silent. It doesn't change the fact that I have the lab until 9:00."

credit 898

FAKING RESULTS USING REAL EXPERIMENTS

You are a pharmaceutical company who has blown through a billion dollars of research and development, only to produce a drug that doesn't work. But you want to get it approved by the FDA* anyway so you can recoup that cost, hapless consumers be damned.**

The FDA requires experimental evidence of the drug's efficacy—a controlled study showing positive results at the 90% statistical confidence level. You need to publish a believable report, even though there isn't a statistically-significant effect. How can you achieve your nefarious goal?

Easy: Run studies repeatedly, until you get a false-positive result. Then publish only that result.

All experiments sometimes give false-positives. An experiment that gives the correct result 95% of the time, still gives the wrong answer 5% of the time. If you run it 15-25 times, you will almost surely hit one of those false results.

This is in fact what pharmaceutical companies used to do to get drugs approved by the FDA. To prevent this behavior, the FDA now requires companies to pre-register their studies and publish all results.***

---

* The United States government agency that approves drugs for sale.

** There are many such examples in the US, such as the decongestant Phenylephrine, with nearly $2B in annual sales, which an FDA advisory panel unanimously concluded[899] is ineffective.[900]

This is also what happens in the social sciences, resulting in The Replication Crisis.[902] An experiment is run once, often with a small number of college students. Occasionally something "interesting" happened, and a journal publishes it. Journals don't wait for other teams to reproduce the result. Nobel Laureates have admitted[903] that this fallacy has debunked their own work, and call for "replication rings" to solve the problem. But scientists are people too, and often prefer the fame generated by an astonishing result to the pain of discovering that they are infamous for perpetuating a false-positive.

Before you shake your finger at them, shake that finger at yourself.

Because you're doing this too.

FAKING YOUR A/B TESTS (UNINTENTIONALLY)

To see how this unfolds, consider this simple example: You're testing whether a coin is fair, i.e. that it comes up heads equally often as tails. Your experiment is to flip it 270 times, measuring how often it comes up heads. Of course we don't expect *exactly* 50% heads, because there's randomness. What is a reasonable range to expect from 270 flips, assuming the coin really is fair? According to the binomial distribution, 90% of the time the result will be between 45% and 55% heads, if the coin is fair.

So, you run the experiment, and you get 57% heads. You conclude the coin is biased, and you say "I'm 90% sure of that." Is this the right conclusion? Probably? Maybe?

*** Although there are plenty of recent cases[901] where companies were allowed to submit only a subset of trials, were approved, and—big surprise!—were later found to be ineffective.

Now imagine you have 10 coins, and you want to test all of them for fairness. So you run the above experiment, once per coin. 9 of the tests result in "fair coin", but one test shows "biased coin".

Should we conclude that the one coin is biased? Almost surely not. Because even if all coins were fair, we know that 10% of the time the test will incorrectly conclude "biased". So this result of `9 / 1` is exactly the result we'd expect if all coins were fair.

But wait a second... what if in fact 9 coins were fair but 1 were biased? Then this is *also* the most likely result! It could have also come up `8 / 2`, but the `9 / 1` result is the most likely.

So: `9 / 1` is the most likely result *both* if all 10 coins are fair *and* if only 9 coins are fair.

So... what exactly can you conclude from the `9 / 1` result? Nothing, yet, not with confidence. What you conclude is that this procedure is insufficient, and that we need to augment the procedure to correct the issue.

The insight is: **You are making exactly this mistake with your A/B tests.**

You are running a bunch of A/B tests. You're looking for (something like) "90% confidence". Mostly the tests have a negative result. Occasionally one works; maybe one out of ten. And you conclude that was a successful test. But this is exactly what we just did with coin-flipping.

In the real world it's often even worse, like using confidence of 85% or 80% and therefore false-positives are much more common.

Or you don't even pick a confidence level. You don't decide how much N you need to make a conclusion. Instead you "run the test until we get a result that looks conclusive."

This particular error of "stopping whenever it looks conclusive" is called "p-hacking" by statisticians, and it's been a well-documented fallacy since the 1950s. The reason it's a fallacy, is that when N is small, random fluctuations will often cause a result that looks like "90% confidence of a positive result," whereas if you continue the experiment, the data shifts back into the territory of "negative result."

I show some fun real-world examples of p-hacking, and counter-examples when the experiment is done properly, in this video (Figure 1).

MARKETERS: THE ACCIDENTAL P-HACKERS

Marketers have been making these p-hacking errors in A/B testing for many years. You are too.

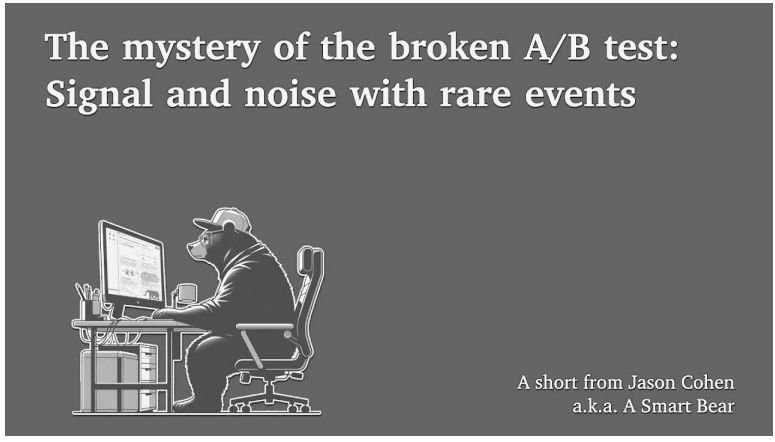


Figure 1: Watch on YouTube[904]

We have data. A study* of more than 2100 real-world A/B tests across Optimizely's customer base found a 40% false positive rate. Marketers never knew it; indeed, the Optimizely software declared the tests "significant"! The Marketers never had a chance.

So, roughly half the time you think "this A/B test was successful," it wasn't.

This explains another phenomenon that you're probably familiar with if you've done a lot of A/B testing:

1. You run tests. Sometimes one is significant. You keep that result and continue testing new variants.
2. You keep repeating this process, keeping the designs that are "better."
3. Over time... one is 10% better. Another is 20% better. Another is 10% better.
4. So, that should be 45% better overall.
5. You look back between now and months ago when you first started all this... and you don't see a 45% improvement! Often, there's no improvement at all.

Why didn't all those improvements add up? Because they were false-positives.

Shopify confirms this theory with years of data.** Their 600-person Growth Team constantly runs funnel-optimizing A/B tests, but because they understand the p-hacking phenomenon, they not only look at short term results (2-8 weeks) but whether the results hold over the long run (12 months). Their results: Of the tests which show statistically-significant improvement in the short-term, 35% of the time the effect disappears over the long run.

And other people experience this anecdotally (Figure 2).

---

* Here is the study,[905] and here is a blog post[906] from the author, addressing concerns and caveats.

** From this 2025 interview[907] of Archie Abrams, Shopify's VP of Product and Head of Growth, starting around the 18-minute mark.

dilemmawor 12/05/24

Oh man. We literally have a hundred different ab tests launched every week in our product. And let's say a third end up "significantly increasing conversion rate". This has been going on for a decade, so with CR being increased by 1% several times per week, we should already be at 100% conversion rate. Or maybe 200%. But we're at 9% and it's steady.

credit908

Figure 2

HOW TO STOP FOOLING YOURSELF

The easiest thing is to run the test again.

If false positives are 1 in 10 at 90% confidence, then you should be able to run a second test, and get the same result.

And don't stop tests early. I know you're excited. Just wait. No p-hacking.

And seek large effects, like double-digit changes in conversion rates. Large effects are unlikely to be caused by randomness; small fluctuations are far more likely to be false-positives. And anyway, large effects actually have an impact on the business, whereas small effects don't. This might mean testing drastic changes instead of incremental ones.

That's it? Almost—**there's a much smarter way to go about this**. And if you want to keep your job even with the rise of AI, you need to be smarter than just running a bunch of variants.

credit[909]

"You won't believe it! Anderson's experiment... the results... they're conclusive! *Conclusive* I tell you!"

FORM A THEORY, TEST THE THEORY, EXTEND THE THEORY

Too often A/B tests are just "throwing shit at the wall." We excuse this behavior by saying "No one knows which headline will work; it's impossible to predict, so we just try things."

Not only is this thoughtless and lazy, it also means **you haven't learned anything, regardless of the result of the test.**

You don't want to be a mindless slinger of random phrases. AI can do that too, and AI isn't a good marketer. Instead, you want to **create validated learning.**

To do this, form a theory, then design experiments to test the theory. Example theories:

1. At this point on the website, visitors are ready to buy, so send them down a purchase funnel with a restricted UX.
2. Here people want to learn more, so talk about options and let them explore features rather than being crammed down a funnel.
3. People are on the fence, so we should be more forceful and confident in our language.
4. People can't see well, especially on mobile devices, so we should have higher contrast colors and less text.
5. Pictures work better than paragraphs, especially since people hate reading and half of them don't speak English natively.
6. People are more likely to click buttons than to click links.
7. People from marketing channel X are more likely to be in a Y state of mind, and to be excited by Z.

Perhaps some theories already popped into your mind. Good! Write those down. Then make designs that would perform better if that theory were true.

It's not "shit on the wall" because this time you have a specific Theory of Customer that your wall-shit is designed to test. And that makes all the difference:

The negative result

Let's say you pick a theory, run a test, and it fails. Is your theory disproved?

Not quite yet. Perhaps your implementation wasn't the best manifestation of the theory. Not extreme enough, or had other issues that

covered up the good effects. If you feel this might be the case, run a new experiment.

But if you're still not getting positive results after a few iterations, you have accumulated evidence that the theory is incorrect. That is called "learning." Which wasn't happening when you "threw shit at the wall." Now you know that you need to invent a new, different theory and test *that*.

How useful, and directed.

The positive result

Suppose you had a positive result. Hooray!

Is the theory proven? No, because you read the first half of the article, so you know that positive A/B tests are often false. So, what do you do?

You lean even further into the theory. Run another test that's even more extreme, or a different form of the same concept.

If the theory is truly correct, that will work again, perhaps even better! If it reverts to nothing, you know it wasn't a real result.

Now you're not fooling yourself. You're finding theories that actually correct. That's what "validated learning" looks like.

Since you've actually learned something, you can extend the theory. What else is probably true? What new designs and text and pictures would leverage those insights even more? Now you might make multiple leaps of improvement, rather than spraying random things on your website.

And you're a smart marketer that AI cannot replace.

Most theories won't be right (or at least not impactful enough to matter). Most tests will come up negative. That's frustrating but it's the truth. Even for world-class testers like Shopify's Growth Team.

You *do* want the truth (p. 657)...
Don't you?

Many thanks to Einar Vollset[910] *for reviewing early drafts.*

Chapter 71:
Double your productivity without more work or stress

"Could you at least burn a candle at *one* end?"

credit 911

Zappos COO Alfred Lin enlightens us[912] on **how to become *37 times* more productive in only one year!** Can it be?

Let's hear him out:

> Make at least one improvement [every day] that makes Zappos better. It sounds daunting, but remember improvements don't have to be dramatic. Think about what it means to improve just 1% per day and build upon that every single day. Doing so has a dramatic effect and will make us 37x better, not 365% (3.65x) better at the end of the year.

At first glance it's inspiring. At second glance it's poppycock.

Being 37x more productive is impossible, and I'll show you why. But along the way it will become clear how **becoming 2-3x more productive might be within reach.**

His math isn't the problem per se. It's true that if you improve 1% each day over the previous day, that's a 1% *compounding* rate. My question is: Is it possible to increase your daily productivity by an entire percent every day?

Well, if you can do only 1 pushup, then I believe that you could improve about 1% per day, and be able to do 37 pushups at a time by the end of the year.

But it continues. The *next* year you continue getting 1% better per day, and by the end of the *second* year you can do 1369 pushups in one day. Yeah right.

Things rarely compound like that, even at slow rates. Or at least, there are other factors that overwhelm that simplistic idea.

But there *is* there a smart way to 2x your productivity. (Once!)

Time for a fun math puzzle. Yeah, I know, "fun" is relative... Okay look if you don't like word problems just take a random guess at the answer. If you're up for the challenge, try to solve it without pen and paper. You know, just to prove your $200,000 in student debt wasn't all for nothing.

credit[913]

Here's the puzzle: You get in your car at home and head out towards your mother's house 60 miles away. (Your mom likes this word problem, I can already tell.) You hit traffic during the first half of the trip, so after 30 miles you've averaged only 30 miles per hour (Figure 1).

Now the traffic opens up and you can go as fast as you want. The question is: **How fast do you have to go during the second half of the trip such that you've averaged 60 mph over the entire trip?**

If you're not using pen and paper, maybe you guessed 90? 120?

Actually it's impossible! To average 60 mph you need to travel the whole 60 miles in a single hour. But it's already been an hour! Even if

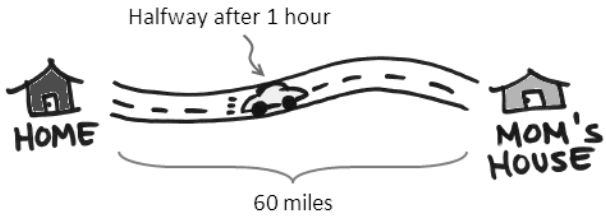


Figure 1

you went 1000 mph during the second half, it would have taken just over an hour to complete the 60 miles, therefore your average is still less than 60 mph.

It's amazing how periods of low velocity wash away gains of high velocity. In the puzzle, if you *doubled* your speed in the second half it would only increase your trip average from 30 to 40 mph. If you *quadrupled* your speed in the second half, your trip average would still be only 48 mph.

This puzzle illustrates the weird math of velocities, and what applies to "miles" per hour also applies to emails per hour or writing code or writing prose or any other "gettin' stuff done" per hour.

The problem with improving your productivity is that so much of your day is occupied by low-velocity activity—dealing with emails you didn't really need to see, dawdling in a meeting that hasn't started yet, or spending too much time reading blogs. (Except this one. This one is a high-velocity activity!)

When half your day moves at 30 mph, it's impossible to make up the time during the other half.

This is one of the problems with Lin's 1% idea—the low-velocity stuff makes it too difficult to improve even 1% overall, at least not every day of the year. Even with 37x improvement in some areas, you still might not be 2x more productive overall.

There's good news here, however! Once you realize that the low-velocity stuff is responsible for most of the drag on your productivity,

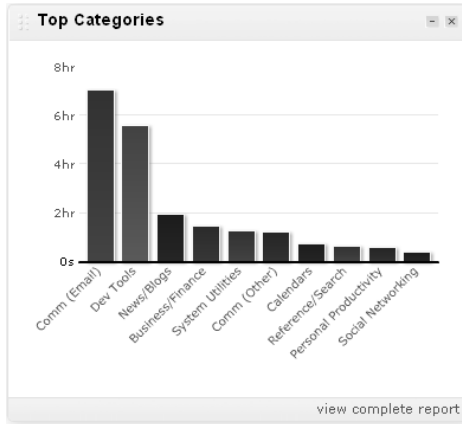


Figure 2

you realize that **the thing to do is eliminate the low-velocity stuff.** Yes it's good to learn to type faster,[914] but cutting down on the time it takes to process useless email[915] might help even more.

Ready for more good news? There are free tools that help you identify what the low-velocity stuff is. I use one called RescueTime.[916] To show you how useful this is, consider this example of my stats for one week (Figure 2).

Whoa—almost eight hours of email. That's a *solid, uninterrupted, full day of nothing but email* I'm blowing through every week. Is that really the way I should be spending my time? Can I ever get to 60 mph like this?

Even the long tail can be instructive. Notice the 45 minutes of "Calendars." A drill-down bears out the awful conclusion—yes I spent almost an hour in Google Calendar. It's true this week was completely packed with events, but being in the calendar app isn't the same as being in the meeting; it means *managing* the meetings. Low-velocity activity.

Another realization: I had an averaged 5.5 hours of activity per day. I was in the office for over 8 hours every one of those days—the

rest is sopped up with meetings, office chatter, and lunch. Here's the mythical eight-hour workday quantified—I'm starting with 5-6 and even then I spent much of it fielding email.

Once you see the numbers it's easy to correct. I now notice more when I'm in an office conversation that's past the point of being productive. There's millions of tips for how to process email more efficiently.[917]

So if you're serious about wanting to increase productivity by, say, 2x, you can. Identify the biggest perpetrators of low-velocity activity and eliminate them.

And makes sure your major "big rock" tasks are the right ones (p. 221).

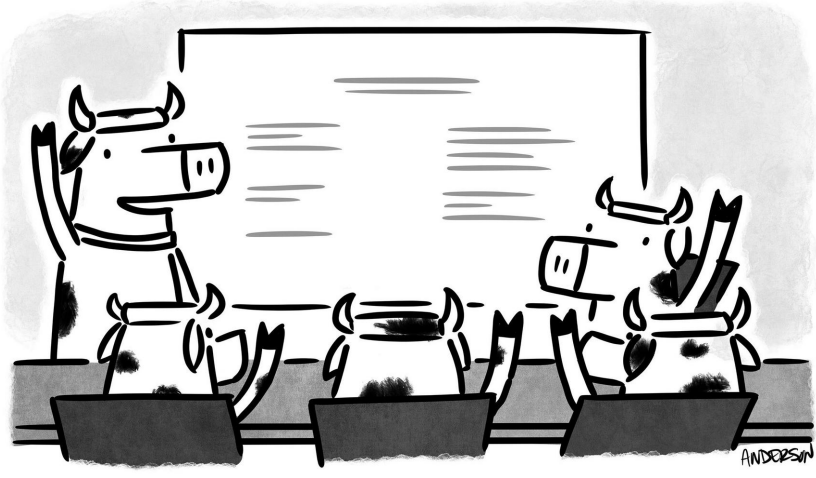
The best part is, none of this means working late or working harder. Just stop averaging down!

credit[918]

"As you can see, we've been able to significantly boost productivity by making everyone work much, much harder."

Chapter 72:
Ignoring the Wisdom of Crowds

JELLYBEANS · MULTIPLE CHOICE · UNIVERSAL?
ANALYSIS · VETOING INNOVATION
FURTHER READING



"Another unanimous vote! Man I love the herd mentality!"

Let's start with some fascinating, unassailable facts. Then we'll assail them.

JELLYBEANS

In 2007 Michael Mauboussin[920] presented a big jar of jelly beans to his seventy-three Columbia Business School students. How many beans did they think it contained?

Guesses ranged from 250 to 4,100; the actual number was 1,116. The average error was 700—a massive 62%—demonstrating that the students were awful estimators.

Now here comes the weird part. Even with all these wildly incorrect guesses, the *average of the guesses* was 1,151—just 3% off the mark. The group's average was closer than almost one person's guess—only 2 of the 73 students guessed better.

So although individually everyone was woefully inaccurate, collectively the group was incredibly accurate.

Was this a fluke? Hardly. The experiment was made famous in 1987 by Jack Treynor.[921] In his case it was 850 jelly beans and 56 students. The group average was only 2.5% off the correct number; only one student guessed better. The study has been repeated many times with similar results.

This eerie effect goes beyond jelly beans; it's also a big help when you're trying to make money on TV.

"If I say 4+4=7 it's wrong. If you say 4+4=7 it's wrong. But if the entire first grade says 4+4=7, well, now she has to take us seriously."

credit[922]

THE BEST MULTIPLE-CHOICE TEST EVER

A contestant on the game show *Who Wants to be a Millionaire* wins a million dollars if she correctly answers fifteen consecutive multiple-choice questions. If she's stumped along the way she has three "lifelines": (1) eliminate two of the four choices, (2) telephone a friend, or (3) poll the audience. The jelly bean experiments imply that this third choice might be pretty good. Is there as much wisdom in the crowd for pop culture and science as there is in counting jelly beans? See for yourself (Figure 1).

The TV studio audience predicts the correct answer an astonishing 91% of the time. Remember, these are questions from all domains of knowledge, all ranges of difficulty, polling a group of people whose only qualification is that they happened spend this weekday afternoon in a TV studio.

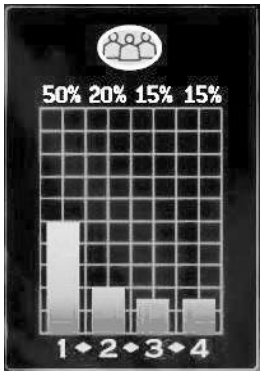


Figure 1

To quantify how amazing that is, compare with the accuracy of the "phone a friend" life-line where the contestant gets 30 seconds with a pre-determined person. This accomplice is probably considered to be "the smartest person I know," plus has access to ~~the web of lies~~ Google and Wikipedia.

The intelligent friend with broadband access to the entirety of human knowledge gets it right only 65% of the time.

Crowd wins again.

IS THE RULE UNIVERSAL?

There's seemingly no end to studies like these, all showing that the crowd is smarter than the individual. Is this a universal rule? Should we be leveraging this power more often?

Big companies do use crowd wisdom. You always hear about advertising campaigns being honed by focus groups of "real people." (I'd like to see the questionnaire that distinguishes "real people" from that elusive other kind of person.)

credit[923]

"It's not cheating, it's crowdsourcing."

However, company messaging, product features, advertising layouts, and the other creative aspects of business require *innovation*, and we know that design-by-committee is the antithesis of innovation. Average products designed for the average consumer[924] is the opposite of innovation, and probably a bad product strategy (p. 317) too.

So what should we do? Can we rely on the wisdom of the collective or should we trust a stroke of inspiration?

ANALYSIS OF HOW "CROWD WISDOM" WORKS

Let's take another look at *Who Wants to be a Millionaire*.

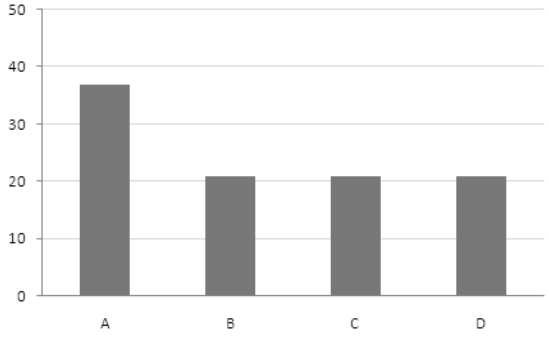


Figure 2

Suppose there are 100 people in the audience and only 16 of them know that `A` is the correct answer. Of the rest, none knows the answer and they vote randomly. The result of the vote will be: 37, 21, 21, 21 (Figure 2).

Oh gee, it's awfully similar to the earlier graphic of a real audience poll.

(For those of you so inclined, it's fun to try more complex scenarios, although you'll find the result is always similar. For instance, what if only 11 know the answer is `A`, 15 each know that `B`, `C`, or `D` are certainly *not* the answer (and vote randomly for the other three), and the remaining 44 have no clue and vote randomly. In this scenario, the vote distribution is exactly the same as the simpler example!)

So we have the interesting result that a mere 16% of the voters were able to make choice `A` the *clear* winner—nearly double the next closest answer. The reason? The ignorant people vote randomly and *their votes cancel out*, leaving the few in control of the result.

THE CROWD VETOES INNOVATION

Now that we understand how crowds can be right, let's see why this same process doesn't work for creative endeavors.

Consider what happens when you're planning a holiday meal. There's a range of fantastic things you could cook, but wait: Some people can't take spicy food, Uncle Bill is allergic to garlic, Aunt Sarah doesn't eat red meat, Timmy doesn't eat anything green, ...

Eventually you realize there's only way to please everyone: Cook something bland, mild, and safe, like chicken and rice (Figure 3). But does chicken and rice actually *please* anyone? Not really, it was just what everyone *hated the least*.

Votes don't converge on something wonderful. Rather, votes are *vetoes*.

Of course if you're a catering company for weddings, chicken and rice might be the way to go! After all, no one goes to weddings for the food, so your primary goal is to piss off as few of the 300 guests as possible. Come to think of it, chicken and rice *does* seem to be popular at those sorts of functions...

But this isn't a good strategy for startups (p. 489). Little companies need a niche—a market space they can completely, unquestionably

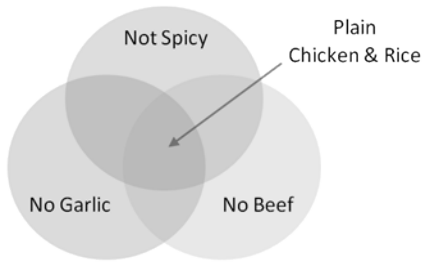


Figure 3

own, not some gray middle-ground where your attempt to offend no one also means exciting no one.

There is "wisdom in the crowd" when there is an objectively-correct answer, and when the errors cancel out, like when estimating jelly beans or answering pop culture questions.

In creative work, votes eliminate the interesting edges, because votes result in subtracting rather than adding, leaving only the boring residue that no one hated enough to vote off the island.

That's not how great products are made.

FURTHER READING

- *The Wisdom of Crowds*[925] by James Surowiecki,[926] with more stories and implications for Wall Street, and his (more expert than my) analysis on the five elements required to form a wise crowd.[927]
- *The Difference*[928] by Scott Page, explaining how diversity makes a group smarter. The inspiration for my *Who Wants to be a Millionaire* example.

Chapter 73:
Human + Fallible = Love; Corporate + Sterile = Refund

"Do me a favor and hit her with some lower expectations first."

A lovely new company/customer etiquette has emerged, and small startups are especially suited for exploiting it, whereas incumbents cannot (p. 295). I hope you're not ignoring it.

Just yesterday someone explained to me what they expect from their website hosting company:

> I want someone else making sure the server doesn't go down. Or, if it does go down, **I want someone to apologize to me.**

Ten years ago, that bold text would have read: "Or, if it does go down, I want **someone to scream at**." Or: "I want someone to **give me a refund**." The new attitude is not "Those assholes better not *ever* screw up," but rather "I expect them to try hard, to care, and to treat me well *when* they inevitably screw up."

This doesn't mean you get a free pass to abuse your customers, then earn forgiveness from a faux-heartfelt "open letter from the CEO." Rather, it means:

- **You're doing your best to do right by your customers**, evidenced continuously through all your communication—blog,[930] tech support (p. 1503), website[931]—not just after a crisis.
- **You're learning from your mistakes**, evidenced by problems tending towards the esoteric, and by explaining in your apology what steps you've taken to avoid this and similar classes of error.
- **You're doing everything in your power to be the best**, evidenced by a culture of awesome employees and inventing new ways to make your customers successful, so mistakes are ordinary human error, not negligence or indifference.

It's not even the apology itself; no one's convinced when a large company issues an insincere, legally-vetted "official apology" that doesn't fix anything. What that initial quote really means is: "I want to work with people who behave like real people, who are obviously

trying their best, and who respond to problems as earnestly and quickly as can be expected."

In short: **People readily forgive sincere human error, but become adversarial and distrustful with the typical, sterile customer/provider relationship.**

There are many ways to earn this love; here's a bunch (p. 275).

The biggest in this context is to do what every blog-about-blogging and tweet-about-tweeting sternly instructs you to do: "Be human." Umm, what? Compared to what, being feline?

(Isn't it weird that we have to be *told* how to "be human?" WTF?)

"No no," they say, "it means let your humanity show—be authentic (p. 627)." Oh brother, ok, how do I do *that*?

The typical advice for "being authentic" is to "just be yourself (p. 1433)," but I don't know what that means either. Thales[932] said the most difficult thing is to "Know Thyself,"* so it must be *really* hard to do that over Twitter and AdWords. (By the way, Thales also said the easiest thing is "To Give Advice." I'll let you bask in the irony for a minute...)

So I suppose one route to "finding your voice" is to take stock of your total life experience together with your ten-year goals,[933] then synthesize a compelling, internally-consistent philosophy, apply that to all your actions and communications, and summarize it in four punchy words on your home page.

Yeah right, who can do that? Not me, I can't even decide what to have for lunch.

So instead, here's a few practical ways to discover what's essential to your personality and point of view:

Criticize others.

If you especially enjoy someone's slogan, why? Is it because it's funny, clever, specific, unwavering, simple, conservative, confident, or ballsy? Conversely if you loathe someone's "About Us" page, why? Is it because

* If you're on this journey, the bottom of this article (p. 399) has a list of questions you can use to try to figure it out.

it's too personal, not personal enough, too detailed, not detailed enough, silly, formal, useless, childish, lengthy, or arrogant? When you see something that strikes a nerve, complete the sentence: "I absolutely [love|hate] that because ...."

Decide what you are *not*
For example, you might say "I hate companies who use formal language; I'm never going to allow formality to dictate how I communicate." Or the opposite: "I hate companies who think it's funny and clever to use informal language; I'm going to instill confidence by showing that we behave like grown-ups." It's easy to identify corporate stuff that pisses you off; use that to decide both what not to do and what to do instead.

Copy something you love.
Sounds weird I know—how can *copying* lead to a unique, personal style? But if you think about *why* you love something—a company, an attitude, a writing style, a philosophy—it's because you identify with it so completely. It *is* you! Of course over time you'll morph that copy into something unique, but there's nothing wrong with getting a head start by imitating something you wish you had thought of yourself. Careful though—I'm not advocating plagiarism! The goal is mimicry, not theft; influence, not carbon-copy. Your mindset should be: "The thing I'm copying is a rough draft that needs extensive editing but whose heart is in the right place." Here's more on learning by copying (p. 977).

Even assuming you successful identify what "being human" means to you, it's still surprisingly difficult to implement because **every strong decision you make will necessarily alienate many people even while it's thrilling others.**

If you adopt an informal style, some people will find it refreshing while others find you untrustworthy. If you're proactive in announcing bugs, some people will reciprocate by gracefully putting up with the problems, while others will be shocked—*shocked!*—and will complain on Twitter that you sell shoddy software. If you admit that the entire company consists of two people, some folks will smile knowing they'll get primo customer service while others will flee because of the low probability you'll still be around next year. If you curse on your

blog, many people will wince and click "Back" but others will laugh and click "Subscribe."

And yet: strong, specific, and honest you must be. Yes it means turning off some people, but the remainder will love you all the more (and make sure their Facebook "friends" know it).

It turns out, this actually wins far more customers than you think, and here's why (p. 317).

What's the alternative—having no persona at all? Then why would anyone get excited about you? Why would they put up with your faults? Why would they tell their friends about you?

Is your goal is to become a soulless corporation? No? Well then, do whatever it takes to be soul*ful.*

Chapter 74:
Your non-linear problem of 90% utilization

"You've got a breakfast meeting, meetings all morning, two lunch meetings, your afternoon meetings, a dinner meeting, and then it looks like you've pencilled in 'self-immolation' after that?"

credit 934

Suppose a web server is running at 50% of its full capacity. Browser traffic doesn't arrive in regular, smooth amounts; it comes in spurts and occasionally large spikes. Because the server is under-utilized, when a spike arrives there are spare resources to handle the increase. If the spike is sufficiently large, performance will degrade, and if larger still, many of the requests will be rejected rather than answered; after all, there's *some* limit past which the server cannot do any additional work.

Now suppose someone looks at a report that shows "50% utilization" and says "Hey now, this is a server, not a person! It costs us the same whether we drive it at 50% utilization or 90% or 99%. So let's get our money's worth and drive it into the 90s!"

What happens? Even normal variations in traffic will drive the server past its capacity. The average time to respond to a request will skyrocket, and often requests will be dropped altogether. Not due to an unusual event, but all the time. **The system is now brittle**—not good for costs, not good for the quality of the product or customer experience—just bad all around.

Maybe we can drive high utilization by having multiple servers work as a team. Suppose we have three servers, all serving traffic for the same website, all at 70% capacity. That sounds like a happy medium between 50% capacity (wasting money) and 90% (brittle). The total amount of utilization is 2.1 servers (3 x 70%), so we're nicely over-powered for traffic spikes.

But what happens when one server runs into problems? Suppose it crashes, or the power in its data center cuts out, or someone else breaks the network with a glut of garbage traffic. The 2 remaining servers now have to deal with 2.1 servers' worth of traffic. Both are at 105% capacity, and we're back to broken and brittle.

This isn't really about servers; it's about you and your teams. It's about how your "busy" life not only diminishes your productivity, but how your whole team is hectic, yet bringing itself to a crawl.

We all have a capacity, whether you want to measure it in hours, in energy, in focussed attention, or if you don't want to measure it at all. Instead of web-requests, we have life-requests, whether those are

inbound emails, Jira tickets, Zendesk tickets, Salesforce leads, requests from co-workers, requests from a friend, or families that need our time and attention even more than they need our paycheck.

90% utilization is causing more failure than you realize, not just in burn-out, but in productivity and output. Of course you'll burn yourself up, sacrificing sleep, health, friends, family, and other things you mistakenly take for granted, but I suppose you knew that already. You're trading that for super-human productivity, right?

But you won't even receive outsized professional gains as a reward. **This condition is a combination of frequent context-switching and interruption—the Twin Enemies of productivity.** Work-completion will drag out because it's constantly interrupted. Some will be abandoned.

Worse, in many organizations *everyone* is operating at 90%, which then reacts like the three-server system, where the inevitable hiccup from any one person causes a ripple effect that hurts several other people or projects. Since *they* are over capacity, rather than absorb the

"I'm sorry, I was sure there was something over there that I needed to bark at. Please continue."

credit 935

spike, they too will ripple the problem to others—a cascade like the run-away chain reaction of an atom bomb.

The key word here is *inevitable*. People get sick or die or leave or change or have to run an errand or want to do even one minor piece of work that wasn't mapped out weeks in advance. True emergencies arise that deserve to interrupt work. This is not something you can "architect out" of the universe; rather, you need to build a system that assumes variation and interruption, and design your personal and team's work-style to be resilient to that variation.

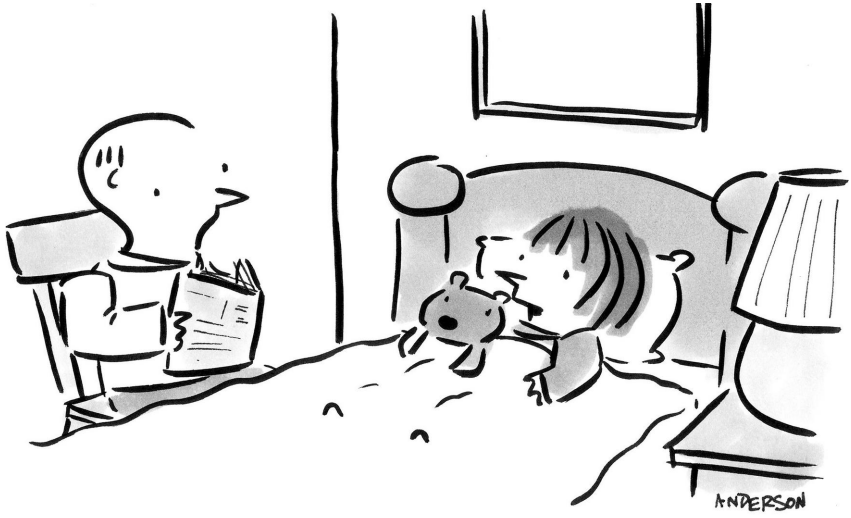
The ideal is probably a situation where most of the time you're in the safe zone, with occasional surges into high gear **for a short period of time** and **for good cause**. For example, a brand new product launch is usually attended by some extra time fixing bugs, especially post-launch where it hits real customers and a few issues are discovered that we all agree should be fixed swiftly before more customers encounter it. Or there could be a clear-and-present danger to the company that requires a special, time-bounded rally. Or you could use infrequent and brief surges in a fun way, like a Hack-a-Thon or a Bug Squash Competition or a Ticket Kill Day.

We're erring on the side of over-utilization, and rather than providing the benefits of competitive advantage through higher productivity, it's creating needless turmoil and lower productivity.

Don't let yourself, or your team, fall into the trap.

Chapter 75:
The “Talk vs Walk” workshop

THE METHOD · TAKING ACTION
COMPETITIVE STRATEGY



“Go back and give that part a little more oomph.
Really make me believe!”

Despite the insipid rhyming, it's a surprisingly useful question to analyze:

You talk the talk, but do you walk the walk?

Whether it's the early-stage company, understandably exaggerating its product's qualities on the home page in a lived experience of "Fake it 'till you make it,"* or the venerable company whose marketing department isn't taking enough credit for legitimately world-class qualities, because they've become so second-nature that it doesn't occur to them to brag about it.

Plotting items on a "talk" vs "walk" chart is not only useful in analyzing current-state, but also in uncovering ways that both the Marketing and Product groups can sell better and create better products.

THE METHOD

Write-storming ideas

Using a whiteboard,** set up a vertical axis called "**Talk**: How much/loudly do we claim this" and a horizontal axis called "**Walk**: How well do we pay this off in practice." (Figure 1)

Then, independently write-storm:*** Create "sticky notes" and place them wherever each person believes they should go on the chart.

---

* Again with the rhyming?

** Likely virtual nowadays, e.g. Miro, LucidChart, or even Google Slides; something where folks can easily create and drag around "cards," and see the results immediately.

*** **Do not brain-storm items one-by-one.** In live brain-storming the conversations follows the fastest and loudest people, rather than leveraging everyone's brains to invent different ideas. Live brainstorming filters out ideas that take longer than a

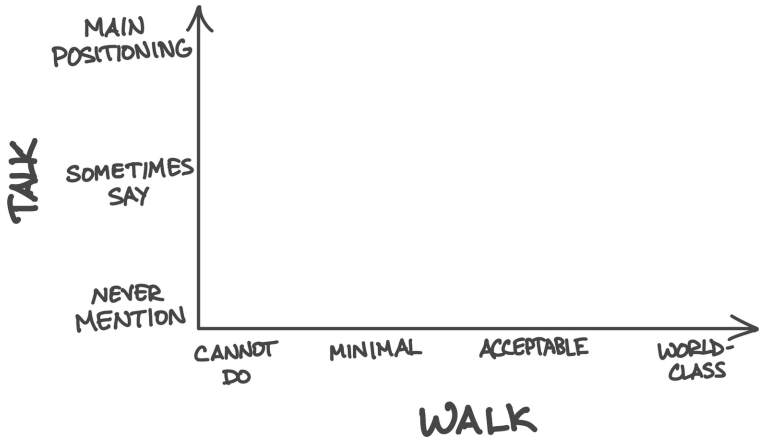


Figure 1

You could even allow several days, giving people time to think and to schedule the thinking during periods when their brains and work-environments are most effective at tackling such projects.

Where do ideas come from? Because the exercise is about "what our company claims," obvious sources are the public website, advertisements, public presentations and webinars, quotes from your employees in the media, sales materials, and support materials like knowledge-base articles or copy/paste shortcuts. If you're quoting directly, put the phrase in double-quotes to emphasize that "we really do say this!"

There are also the things you're doing that you're *not* talking about. Those things are more difficult to generate. Sources are what you're maintaining (i.e. existing systems owned by some team), work from recent sprints (i.e. work you're actually doing), long-term planning

---

few minutes to think of. Live brainstorming eliminates the possibility of having an "ah-ha" moment in the shower. "There is not a single published study in which a face-to-face brainstorming group outperforms a [write-storming] group" observes Leigh Thompson in *Creative Conspiracy*[937]. Recent research[938] suggests that people are less creative in generating ideas over video-conferencing versus in-person; write-storming restores the creativity because it's independent, even if the discussion is over Zoom.

documents (i.e. larger initiatives), and prompts like "What are we really good at?" and "What technology assets or human capabilities have we built up over time?"

The hardest things to think of, are the things you're not talking about and also not doing. Isn't that an infinite number of things? Yes, but it's useful to list things you reasonably *could* say or do, but are intentionally avoiding. One source is backlogs of features requests from customers, support, sales, marketing, product, and engineers; whether you plan to do them eventually or not, they are "items of interest." Another source is key, differentiating things that competitors are saying or doing (whether you're avoiding those things purposefully or reluctantly). Yet another is your own long-term plans for big initiatives you haven't gotten to yet.

Debating placement is one-third the value

Come together as a group, take each idea in turn, and debate where it should go on the chart. Coalesce duplicates into a single card; if the duplicates were originally strewn all over the chart, that's even more indication that a discussion is useful. If there's immediate widespread agreement on the right location for a card, skip the discussion to save time.

Generally you'll find three sources of disagreements. All three are useful, but in different ways:

Lack of knowledge
"We don't talk about this." "Yes we do, it's in all the sales materials!" "In support, we tell people the opposite." "We say this in sales but not on the website." "We *used* to say that, but not since the latest website revamp."

This is a great excuse to get everyone up to speed about what we're saying, and where. If the answer varies (e.g. "we say this a lot on the home page, but nowhere else"), you can place the item more in the

middle of the chart, but add a "comment" to record that detail; it will be useful later when you decide how to act on this information.

Disagreement on terminology
People often interpret the same phrase differently. If you can't agree on what it means internally, imagine how confused customers are! Or a new employee who isn't part of the echo chamber?

An example will illustrate how this is useful: In a recent exercise at WP Engine,[939] multiple people put "Enterprise-Grade Security" on the board. We talk about this constantly, so no one disagreed that it was high on the "talk" axis. But there was disagreement on "walk," ranging from the middle to the far-right.

But how is this controversial? We're well-known for having the best security in the industry. Our customers include some of the largest security companies and financial institutions in the world, both of which maintain the highest bar for security; we are certified with multiple accreditations; we leverage myriad security products covering every layer of the tech stack; we commission third-party pen-tests for every new product and periodically for all products; we conduct annual company-wide training to guard against social engineering attacks; we have formal written security policies that we actually enforce. With all this, how could you not agree "Security" is 100% on "walking the walk?"

Turned out, the person who placed it in the middle was specifically thinking about a few items in the backlog that would make a certain corner the platform even more secure. We'd never experienced an incident because of, and there was no reason to believe we would. Still, this person thought, If there's still a few obvious items in the backlog that we *could* do, then we're not yet fully "walking the walk" on security.

A discussion resolved the difference of opinion. We decided that "great security" isn't defined by never having a ticket in the backlog; after all, nearly all software updates contains security patches (check the version history!). That doesn't mean the software wasn't already secure. Indeed, it's the opposite: Being so diligent that we even "fix" things that aren't broken, thinking a few steps ahead, proactively preventing incidents, *is*

what *makes* us secure; it's a sign that we are in fact "walking the walk."

The group finally agreed that the ticket belongs on the far right side. The realization, however, was helpful both for the engineers who have a better mental model of "what security means," and for the marketers who have more stories to tell about how excellent our security is.

Disagreement on target persona
Sometimes you realize that a subset of your customers would agree that you're "walking the walk," but others would disagree.

For example, perhaps the performance of your product is excellent for most customers, so when you say "fast user interface that never gets in your way," most people would agree. But maybe 2% of your customers have huge amount of data that slows your UI to a crawl. Furthermore, that 2% are important to you—they might be your reference customers, or the ones paying you the most money.

What do you do in these fractured circumstances? A simple technique is to place the item in a sort of "weighted" location. So e.g. if 80% of your customers (by revenue) would consider this completely "walking the walk," but 20% would say it's pretty low, perhaps you place the item at 80%. However, add a comment to the sticky-note explaining this detail; it will be useful later when you decide what actions you want to take.

Another technique is to split the item. So e.g. "Fast UI (for customers <100 properties)" and also "Fast UI (for customers with ≥100 properties)." Then place them independently.

If you have different, distinct target personas, you might find a lot of the cards are being split. In this case, consider making two different charts altogether, i.e. one for "Talk vs Walk for Persona A" and the other "Talk vs Walk for Persona B." This is easier to read and to act on.

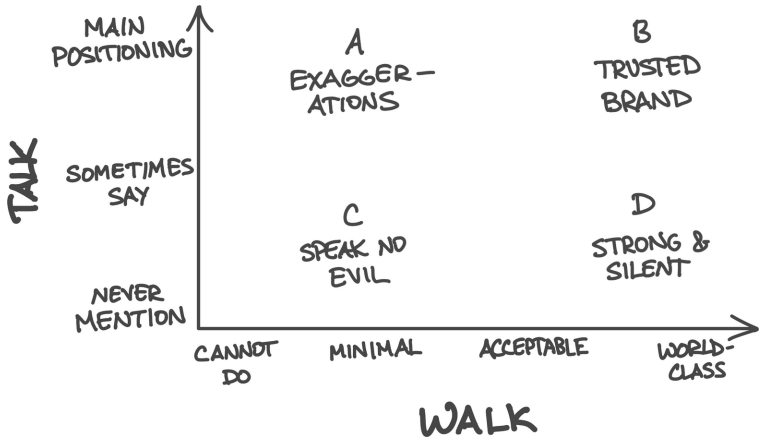


Figure 2

TAKING ACTION

Just having these discussions is valuable, but most of the value is in doing something with the resulting chart.

In general the question is: Is it useful to take action to shift cards in various directions? It's useful to consider this by quadrant (Figure 2), because those items have different *status quo* interpretations.

Quadrant A: Exaggerations

These are the original source of the line "you talk the talk but you don't walk the walk." When you're "faking it 'till you make it," but still just faking it. What to do?

Do nothing (ethically)
We understand our claims are tenuous. We should ensure that our exaggeration is not unethical, e.g. claiming a medical benefit that does not exist, or charging for a feature that does not work.

Live up to your own expectations (shift →)
This message is resonating, and is a critical to our positioning. But if it's so critical, we have to back it up with reality! We'll build and ship specific things in the next six months that will allow us to shift these sticky-notes rightward to meet our customers' expectations.

Don't write checks you can't cash (shift ↓)
We're attracting the wrong sort of customers—people who expect something we don't deliver. As a result they waste time in customer service and then cancel anyway—a net loss for us, and a terrible experience for them. We're not even getting useful product feedback, because we're interviewing people who aren't our target persona. Let's stop saying these things.

Quadrant B: The trusted brand

Trust is earned by repeatedly keeping your promises; that's what's happening in this quadrant. This is your brand, because it's what you say and also what customers actually experience. This is a strength, because everything is constructively aligned. What to do?

Do nothing
We're doing well here, but doing incrementally better will not win us more customers (i.e. Kano[940] "must-be" or "indifferent"). Therefore, we should track our performance on this item as a KPI, but rather than continue investing to maximize it, we should only promise that it won't fall below some minimum threshold of acceptability. This ensures we won't unintentionally lose this advantage. Instead, focus our attention on accumulating more strengths, or mitigating some issue.

Leverage strength for sales (shift ↑)
Can we make even more fuss about how awesome we are? More confident competitive positioning, leveraged in advertising, more content marketing on the subject, emphasized more in sales processes, highlighted more in the product UI. We might even be able to make sales by emphasizing one strength above all others.[941]

Double-down on strength (shift →)
It's often wise to invest more in what's already working, especially early in a product's life when it's hard to have even one thing that is special, or a reason to buy despite its many deficiencies (p. 1271). This only works when customers value the additional investment (i.e. Kano[942] "attractive" or "performance"), but if that's the case, it might be the most strategic thing you can do.

Pivot (shift ↓)
While it's unusual to jettison a strength, it makes sense if the company is pivoting: Changing its target, or how its brand and product is positioned against the competition.

Quadrant C: Speak no evil

Often there's not much in this quadrant, because it's easier to think of things you're saying and doing, and these are neither. However, it's valuable to list "things we know are weaknesses" or "things we're saying 'no' to," especially if it's a common topic of conversation at the office, or if a competitor is successfully talking and walking that item.

Do nothing
This is the correct answer for nearly everything here. It's expensive and high-risk to turn a weakness into a strength; your energy is probably better spend executing on your current promises and doubling-down on strengths.

Turn it around (shift →)
If this is truly a key area that is critical to succeed in the market—which is another way of saying "strategic"—then this is a weakness to highlight, and to invest in. Should you shift it up (start talking publicly) before you've shifted it right (pay off the claims)? Probably not: You tip your hand to competitors, allowing them to react before you can defend. You also set incorrect expectations for customers; later when you really *can* pay off your words, you've already trained them to disbelieve you. Conversely, talking first can freeze the market,* but only if your market

leadership position is so strong that a press release actually does change market behavior. Unlikely.

Quadrant D: The strong, silent type

If you are excellent, why not get credit for it?

Do nothing
There might be a good reason for not making a fuss. It could be a trade secret that you don't want competitors knowing about (but you could brag that a trade secret is part of the magic that makes you different). It could be "boring" internal operational stuff (but customers might be comforted knowing that you're operationally advanced, and a tech blog is a recruiting tool).

Talk about it (shift ↑)
Even if this is a secondary or tertiary message, it's nice to arm sales with more material, or test it in advertising, or make a landing page for the subset of customers who would be impressed.

APPLICATION TO COMPETITIVE STRATEGY

Turn the tables on your competition to reveal more insights.

---

* "The U.S. District Court ... suggested barring Microsoft from making vaporware announcements because doing so can allegedly freeze the market and discourage buyers from purchasing competing products. ... 'I feel very sure there have been many times when Microsoft has announced products to freeze the market, [but] lots of companies have done that,' Kertzman said."—Computer World, 1995[943]

credit944

Adding the competition to both charts

Build the Talk/Walk chart for a competitor.

Place a copy of your cards onto their board. There might be a lot in quadrant C (“Speak no Evil”).

Then create the new cards that makes sense for them. These prompts help generate ideas:

- What do they emphasize (big or bold font) on their home page, “pricing” page, and “about us” page?
- What main features do they highlight on their “features” page?
- On what points do you lose to them in sales? Which are they legitimately great at, versus which are they just really good at selling?
- On what points do you beat them in sales? For which do they deserve it, versus which are you just good at claiming?
- What do their customers say about them (e.g. on Twitter or testimonials)?

- What do your customers say about them (e.g. when complaining or asking for features, do they reference the competition?)
- If you were making an investor pitch on their behalf, what would you say? How much of that would be true, versus aspirational?
- What cultural or moral principles do they or their founder or their CEO publicly espouse?

Whatever new cards you add to their chart, also add to your own chart, even if it mostly creates a pile in quadrant C. This is where it gets interesting.

Analyzing the competitor's strengths

It's especially instructive to compare your competitor's strengths to everything on your board. Place their strengths onto your board in a unique color, i.e. evaluating whether you also talk about those things or not, or you also do those things or not. Every quadrant is actionable:

Quadrant A: Exaggerating against their strength
You're both making a claim, but only your competitor is fulfilling that promise. Over time this erodes your brand; potential customers start dismissing you, saying "Yeah, everyone claims to have X, but only P really does X." You were already claiming something ahead of having it, and now the competitor is exposing your little fibs.

Before you decide to knuckle down and match the competitor strength-for-strength, critically consider whether you want to run a race where you're starting out behind. Do you have what it takes in long-term will-power, and investment in time and skill, not only to catch up to where they are today, but to where they will be in another year? Will the result of that expensive and risky activity pay off enough to be worth it? Occasionally the answer is "yes," but usually the answer should be "no, let's move this card downward—stop making this claim—and instead focus on increasing our differentiating advantages in Quadrant B, where *we* are the ones in the lead."

Quadrant B: Table-stakes; find the differentiation
You're both doing well here. Perhaps these topics are considered "table-stakes" for any competitor in the market. That's not a problem, but neither is it an advantage for you. Table-stakes are important to maintain, but it's a mistake to over-invest in them at the expense of things that differentiate you, or at the expense of the few key weaknesses that you're converting into strengths.

Your comparative advantages are the cards in this quadrant that *aren't* shared with the competitor. Do you have a significant set of those cards? If not, inventing some might be the single most strategic thing you could do.

Quadrant C: It's not your thing, and that's OK
This is where your competitor is differentiated against you. It's easy to get discouraged or think this is a problem; it's not. The only problem is if you don't have your own unique strengths (p. 891). If you both have your own strengths, that's just defines your respective niches, which is useful to recognize.

As with Quadrant A, it's highly unlikely that you should attempt to "turn it around" and become strong where your competitor is already strong. But, if there's one item that's truly important to win, you could set that intention. Just know that you can't have more than one or two of those at a time, and often you'll only get to mediocre, not a strength. It surely must be treated as a Rock (p. 221).

Quadrant D: Don't give them a free pass
You're letting your competitor take all the credit, even though you're just as good. Change that! Sure, you'll probably just match them, not create a new differentiated advantage, but since you've already done the work to be excellent in this area, don't give them an uncontested win in sales calls.

I hope this exercise is as useful to you as it has been for us!

Chapter 76:
The Serengeti Plain: Fallacies that aren't fallacies

FALLACY OF FALLACIES · ANTI-FALLACIES

"What do you have that's sick or injured?"

I never want to hear this narrative again:

> **Back when humans were living in the...**
> Serengeti Plain *or* Saharan Desert *or* coming down from the trees...
>
> **it made sense for us to...**
> do something the self-satisfied speaker scoffs at but graciously forgives a primitive species for doing.
>
> **This worked because we were just trying not to get eaten by a**
> lion *or* tiger; by the way there are no tigers in Africa, but don't let facts get in your way!
>
> **But, in our modern society, this is a fallacy! That's why you have to...**
> embrace buzzwords (p. 627) like "first principles" and "maximize expected value" and "Bayesian priors"...
>
> **to transcend your stupid "lizard brain" and be...**
> a frontal lobe user, like the oh-so-enlightened speaker.

You don't know what it was like in [the only African region you can recall from David Attenborough shows], you don't know how genetic pressures[946] actually work, you can't ignore the hundreds of generations since the rise of civilization, and **not all heuristics are "fallacies."**

"Fallacy" implies it's dumb because it runs against cold, hard, scintillating, pure, perfect logic.

But I don't agree with some of that perfect logic.

WHAT IF "EXPECTED VALUE" IS THE FALLACY?

One of my pet peeves is the idea that we should always "maximize expected value," which I believe is rarely the correct way to make a decision (p. 867), but always generates a tsunami of arguments on Twitter.

Economists claim that "maximizing expected value" is what logical people do. It doesn't bother those same economists that they cannot predict any major metrics of the economy[947] or markets (p. 193), while constantly issuing memos excusing their models for getting it wrong for the 74th time in a row. All while calling everyone *else* irrational.

Let's see why "expected value" might be the fallacy:

I invite you to play a game. The game is, we flip a fair coin. If it comes up heads, I will double your life savings. If it comes up tails, you lose all of your life savings. In other words, it's like going all-in in poker with a 50/50 chance, except this is your actual life, not however many chips you brought into the casino. This is for everything.

How many people want to play that game? If you have very little in the bank, you might want to play, because it makes little difference either way. But among people who have spent years socking away a nest egg, few would take that chance.

From an "expected value" perspective, it doesn't matter whether you play the game or not. The expected value is the same—zero.* But

* The definition of "expected value" is the sum of all the outcomes weighted by the probability of each outcome. In this case, if your savings is `s`, the expected value is `-s*0.5 + s*0.5 = 0`

of course, it *does* matter whether you play the game, and different types of people will want to or not want to play the game, which *proves* that it matters. And I don't think there's anybody on Earth who thinks that characterizing this game as being a "zero" is an accurate or useful characterization.

Now let's really mess with the expected value acolytes. The game is the same, except the probability is 55% that you double your life savings and 45% that you lose everything. Does that change whether you want to play the game?

Very few people would change their mind based on this minor alteration in the game's rules. Losing all of your life savings at 45% is essentially the same as at 50%, and if it was bad at 50%, it's still bad.

But now the expected value of this game is positive.* So the "maximize expected value" people would say you're being illogical if you opt out of the game.

In particular, you would say you are succumbing to the fallacy of "Loss Aversion." This is the notion that we hate losses more than we love gains, and we're allowing this fallacy to drive the wrong decision of not maximizing our expected value.

But I beg to differ, and you probably do too. I don't think loss aversion is illogical, and I don't think making someone play this game because of some religious adherence to expected value is wise. In fact, I think expected value isn't the right way to think about the game at all. Indeed, I would say those people are succumbing to the "Expected Value Fallacy," which is you are using statistics that apply only if you were to play the game a million of times, and where you're allowed to keep the average result, and yet applying that to one instance of playing the game, which is just wrong.

---

* `EV = -s*0.45 + s*0.55 = s*0.1`, so for example if our life savings is $100,000, the expect value of the game is $10,000.

So, what follows is a set of so-called logical fallacies that I don't agree are necessarily fallacies, starting with Loss Aversion that we just demonstrated.

And none of it is because we used to run away from lions.

ANTI-FALLACIES

Loss Aversion[948]

Valuing the pain of loss greater than the joy of gain.

It took a lifetime of pain (p. 737) and sacrifice (p. 1547) and luck (p. 1035) to accumulate what you have. It is not irrational to be much more protective of losing it than you are greedy about getting more. Especially if the loss is catastrophic, as opposed to an experiment with 1/1000th of your money where the maximum loss is immaterial. Of course, in that case you're happy to play, proving again that the so-called "fallacy" is only invoked when the loss is important.

Endowment effect[949]

Valuing an object more, only because you possess it.

We are emotionally attached to things that we own. It makes sense: There's an emotional investment in having made the decision, in shifting or cementing your identify as "a person who would have this object," and how you believe that ownership will be perceived by others. All of this is real, tangible value.

You also presumably value the object more than its face-value. If you buy a ticket to a concert for $200, studies show[950] you wouldn't sell it for less than $300-$400. Is this because you erroneously value it higher because you "possess it," or is this because the reason you bought it in the first place is that you value *going to the show*, so selling the ticket also means *selling the experience*, which is definitionally worth more than the face value of the ticket.

Over time, we can also grow sentimental attachments; economic theory is incorrect if it asserts that sentiment has no value.

Sunk Cost Fallacy[951]

Continuing an endeavor because of previously invested resources.

In long-term projects, commitment can lead to eventual success whereas "fail fast" actually ensures failure. The emotional and financial investments drive perseverance even when there's little apparent hope (p. 429), which can be the key to overcoming obstacles and achieving long-term goals. Sometimes we pretend we're so smart in avoiding so-

"If two negatives make a positive how come two wrongs don't make a right?"

credit[952]

called "sunk cost fallacy," when really we're justifying bailing out when times got tough.

Availability Bias[953]

Overestimating the importance of information that is readily available.

In startups—and even in scaled-up companies—we often lack comprehensive or statistically-significant data. Extensive research is impossible or impractical. Instead of getting mired in analysis paralysis, scared to act on the data we have at hand, we must act with what's in front of us, and under conditions of uncertainty. This is not a fallacy but a practical necessity, and even allows us to move quickly and

adapt. There are specific strategies (p. 193) for operating like this. It's not a bias, it's life.

Confirmation Bias[954]

Favoring information that confirms existing beliefs.

Constant self-doubt leads to inaction; frequent changes in plans leads to confusion. A leadership team who is always shifting priorities will confound the whole organization. A strategy that constantly changes, cannot be executed.

It's also true that a strategy that never changes is wrong. There is a time to reevaluate plans and strategies (p. 1065), but that time is not "always."

Overconfidence Bias[955]

Having excessive confidence in one's abilities or judgments.

93% of drivers believe[956] they are above-average drivers. While mathematically they are incorrect, this also gives them the confidence to act. Constantly second-guessing each driving decision would make them even worse.

Founders must be overconfident. The most likely outcome is failure,[957] so you have to be overconfident to remain optimistic. At the beginning, both successes and failures look the same (p. 159), so you have to be overconfident to push through. Before Product/Market Fit, you can't know whether or when you're going to hit Product/Market Fit (p. 335), so you have to be overconfident to keep trying things, treating everything as an experiment, not a failure (p. 1261).

Overconfidence is not the same as being blind. Being confident in the vision, but skeptical about every detail, is how you find the truth.

Recency Bias[958]

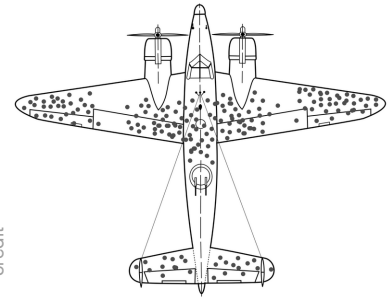
Giving disproportionate weight to recent events.

Early in a startup's life, almost everything is being done incorrectly. Quickly reacting to what's in front of you is one of the ways to iterate to doing things right. "This is how we did it at my last company" doesn't matter, when the last company was 400x larger and 50x older and in a different industry. One of the few advantages a start-

up has (p. 295) is agility, and reacting to the latest information is the definition of agile.

Survivorship Bias[959]

Focusing on successes even if they were due to luck, or share characteristics with failures.



credit[960]

I've been pointing out the problem of Survivor Bias in business advice (p. 449) for more than 15 years. That said, it's not true that "you learn more from failure than from success." From failure you see what didn't work, but that doesn't point the way to what does work.

Yes, successes are lucky (p. 1035), not just good. Yes, companies that failed often do similar things to companies that succeed, which suggests that those things didn't "cause" success. Yes, often there's just one or two most important things (p. 1271) that caused the success, *despite* everything else they did, not *because* of it.

But which is more likely to work: Copying everything about a success, or trying to avoid everything about a failure? Of course the success, because not all of it was luck (p. 1523).

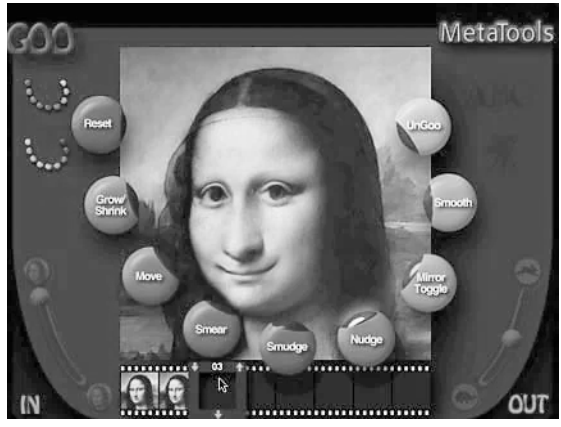
Herd Behavior[961]

People tend to follow the actions of a larger group.

Your product and company should be different in some way that your target customers (p. 317) believe makes you the best in the world (for them).

But in every *other* way, surprises are bad. I don't want to have to calculate the bottom-line price in a unique pricing model, or decode the bizarre controls you created because you didn't want to use menubars, or decipher non-standard icons for things like "copy" and "paste" because you wanted to be creative.

Adhering to norms provides safety, understanding, control, and ease. Those are all desirable qualities in products and companies, except in the very few places where you make a genuine improvement.



Kai's Power Goo. What does "UnGoo" do?

Status Quo Bias[962]

Preferring things to stay the same rather than change.

Stability and consistency foster reliability and trust. A startup thrusts constant change upon its denizens; if you can keep some things constant and reliable, it gives everyone something to hold on to. If you have put thought and effort into your decisions, then it should take even more thought and effort to change that decision; it's not a bias to honor our former selves, so long as new information hasn't come to light that would have changed the those decisions.

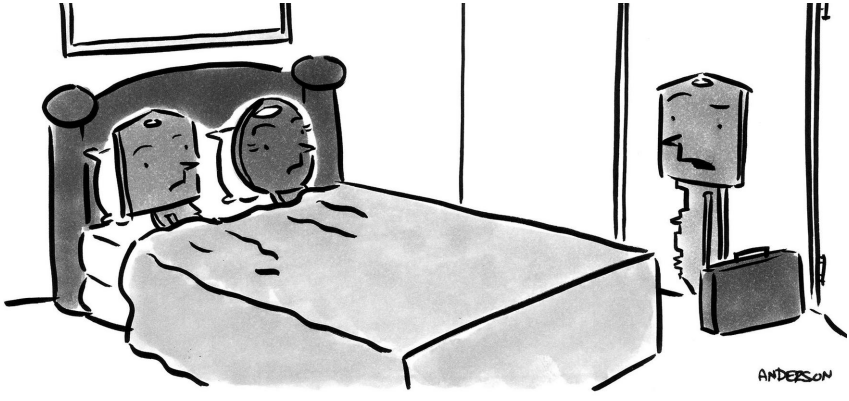
You know, back when humans were living in Statistics for Economists class in college, it made sense for us to pretend that the real world was an idealized environment with only two variables, no complex dynamics, and perfectly rational actors. This worked because we were just trying not to get eaten by a pedantic professor.

But, in our modern society, this is a fallacy! That's why you have to use ideas and tools that make sense for each individual in their circumstances, thinking for yourself instead of parroting phrases off the internet you don't really understand, running your own experiments instead of believing models and theories with more exceptions than examples, to transcend your lizard brain and be enlightened and strategic.

credit963

"It's the gazelles. They got a restraining order."

Chapter 77:
Learn by Copy



"My wife! My duplicate!"

High School English III doesn't teach you how to write prose in the real world any more than a college CS degree teaches you how to write code in the real world. English III teaches you how to write things that English III teachers want to read.

I had to write an essay about Eudora Welty's[965] short story *Why I live at the P.O.*[966] If you haven't read it, don't fret. (Yes, the email program "Eudora" is named after Welty.)

No one in the class understood what the story meant. It just seemed extraordinarily dull where nothing was happening, and then nothing happened, and then it ended. The teacher asked us what it meant; no one volunteered an answer. I piped up: "I really don't know. I'd like to know but I don't." The teacher wasn't happy. The teacher's pet said the same. The teacher was so disgusted with our lack of insight that she never did tell us the answer.

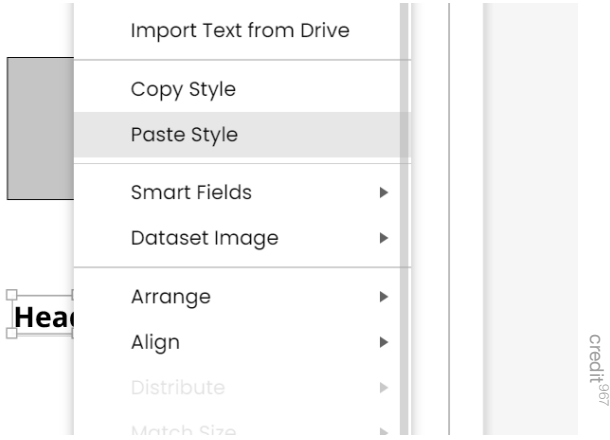
Since I never learned to write well—persuasively, with entertainment, with interest—**I learned by copying.**

Not plagiarizing—that's when you use someone else's words. Rather, I tried to copy *style*. By copying I learned what I liked, what I had some ability to do.

For example, in Hello, I'm 1074018628,[968] a little ditty from 2008, I copied Seth Godin's style.[969] It's 160 words. It makes a simple, solitary point. It runs right from specific example to overarching lesson bordering on a morality tale. At the end you're left inspired to improve yourself, and by extension improve the world.

Except... when you think about how to *implement* your newfound inspiration, you have no tools. The example is apt, but then again you're not *that* bad—you don't mail-merge numbers in MailChimp. And when you're trying to be creative and innovative, you realize that "not shitty" isn't a goal. So then you re-read the post looking for guidance for *how* to be extraordinary, but saying "be the opposite of shitty" is not a lesson at all. It's just Seth (or me) staring back at you like my English III teacher, **full of expectation and no answers.**

So here's what I learned by copying Seth: I like evocative examples, but I don't like leaving the reader without a framework for finding a



Pasting style, not text.

solution. I can't explore an interesting thought in only 160 words. I like trying to leave the reader on an up-note. And I really did *learn*; see for example this article (p. 1433) which exemplifies each of those specific lessons.

I copied others too, sometimes with attribution, like this satirical piece about Joel Spolsky (p. 1553) whose construction was 100% lifted off the first essay of Steve Almond's Rants, Exploits, and Obsessions (not that you asked).[970] Here I learned that it was really fun, but inventing structures like that is beyond my ability.

Even now I'm copying! Even now, after I seem to have "found my voice," with 60,000 subscribers in apparent agreement, I'm still experimenting with other voices. Like this very post, which I'm doing in the style of James Altucher,[971] except without the sexual exploits,[972] and therefore far less entertaining.

Specifically, like this post of James's,[973] I started with a personal story, but it won't quite finish until the end where it wraps up with an almost trivial conclusion but leaves you with a sense of the larger point. I'm using shorter sentences than is my wont. I'm admitting embarrassing or disdainful things about myself, like how I'm insecure and

how I have to copy others' style rather than being strong and perceptive enough to develop my own.

Now that you know this is an attempt at style duplication, you can see for yourself how much better James is at it than I. But "being as good as James at James' style" is not the goal. The goal is to learn and grow as a writer, and this was fun to do.

This technique is useful in almost any pursuit. I was the intermural racquetball champion at the University of Texas, and although it required lots of practice, I *learned* by watching the good players and trying to copy their movements—their swing, where they'd run, where they'd place the ball.

Did any great chef not, at various points in her career, intentionally learn various styles from various cultures?

Even Picasso—one of the most creative, innovating artists in history—first learned his craft by copying (p. 1581) the styles of the Great Masters.

As a culture, at least in America, we cherish creativity, we reward uniqueness. We're obsessed with "innovation (p. 1509)." We're trained that copying is evil.

Rather, copying is one of the best ways of learning, growing, evaluating, and exploring. It's a valid tool so long as we regard it as a means to an end.

Copying ironically helps you discover yourself: The way to discovering who you are, what you value, what you're good at, what you believe, is to try things on, like clothes, to see what fits. Bits and pieces of you are already reflected in the things around you, so pick up handfuls of other creations and see which bits you want to keep.

Eventually the teacher's pet found some commentary on Eudora Welty in our textbook, saying that Welty encapsulates Southern life and mores. We memorized it, in case it was on the test.

Chapter 78:
Scaling by “delegation” isn’t good enough

Founding a company is a selfish act. It will consume every waking moment for the next 1-10 years. It's an act of defiance and irreverence towards competitors and the status quo. This matches the life of a 20-something—fueled by the energy of youth, too young to be jaded, without financial or social dependents. Not all selfish acts are bad ones!

Young founders may fancy themselves wizards of coding (p. 635), design (p. 853), and salesmanship (p. 737); I did! Maybe they're even right, or right enough (p. 1271).

But those skills don't help them build a team of 100 engineers that balance quality with speed, or assemble and manage an international sales team guided by principles other than overwhelming exuberance, or develop a consistent global brand with a voice and adherents, or manage cash flows once the P&L becomes abbreviated "in millions."

Introspective young founders appreciate this, and often the stated solution is "delegation," as defined by:

1. First I'll do it myself.
2. Then I'll understand it.
3. Then I'll have the experience to hire and instruct a new person to do it the right way (i.e. my way)

This is how I did it when I was young and naïve, and I see the pattern repeated all the time.

And it's wrong.

The trouble with this form of delegation is **it results in a team that is not materially better than the founder**, at anything. Which is incredibly limiting for the company, and sadly quite common.

It's a variant of the rule that if you think a certain position at the company isn't useful, it's because you've never worked with greatness[975] at that position. When you're looking for someone who knows what you know, and works like you work, you're not finding greatness, you're finding a substitute for your already-not-world-class performance, and of course you'll get exactly that.

Whereas, as the job of the founder—or any manager—is the opposite: **To hire people who are better than you at every position**, because only then is your organization increasing its strength and abilities.

If you hire and then micromanage (p. 413), or constantly have to "fix" what others are doing, you're a bad manager. Because if the person is in fact not capable, then you hired the wrong person. Or if the person *is* capable, then you're getting in their way and wasting your time. If the person isn't better than you at the position, you're not making the organization better. If the person is better than you, then your micromanagement is destroying that improvement while also destroying their morale and your schedule.

In every case, it's bad management, and it's your fault.

> "It doesn't make sense to hire smart people and then tell them what to do.
>
> We hire smart people so they can tell us what to do."
> —Steve Jobs

This mistake compounds when you're building a larger organization, because then the goal stretches beyond individual excellence: Your job is to build *teams which themselves grow and create greatness.* This is a meta or recursive problem: Not the founder attracting, identifying, and retaining greatness, but the founder building teams who themselves are doing that. This is the best definition of "team-building."

Delegation isn't team-building, and thus it doesn't lead to scale, nor to greatness.

Rather, hiring "up" creates greatness, and the space for scale.

Scaling your business requires that you **convert your initial selfishness into the empowerment of others**. "Delegation" means you still own it but someone else does the work. "Team-building" means the team is trusted to own it, has obligations for that, can figure out

and execute all the details, and is responsible not just for meeting initial expectations, but increasing their expectations of themselves.

You're still in charge, but the team is in command (p. 413).

This is where you achieve true scale in a company. Delegation is where you assign away lesser jobs so you can be even more heroic, because it's still about you. But you're still the bottleneck even if you've made that neck a little wider. Team-building means no bottleneck because the team can be as wide as needed. In fact the best teams measure their own necks and decide how and when to widen further.

This is where you derisk the company by moving from brittle to resilient. Through delegation alone, if one person gets sick, a deadline is missed. Or if someone leaves the company, a strategy isn't executed. With team-building, you have group knowledge. Someone being sick or leaving the company gets baked into the plan.

The moment where you truly understand and embrace this concept is when you can turn the gun on yourself and realize that no one is exempt from this rule. It's relatively easy for a technical founder to agree that she isn't the best person to build a global sales organization, but is she ready to agree that *even where she is excellent*, it's still her job to find people who are even more excellent, not just at individual tasks but at building entire teams?

That's what I did at WP Engine, multiple times, including no longer being the CEO. Here's exactly how (p. 399) I figured out why that was the correct decision.

But doesn't this mean that ultimately **leaders are managing a set of people, all of whom are better-qualified than that leader to do those jobs**? And isn't that difficult to manage, after all how do you argue with those people, and how will you earn the respect and confidence of those people? Yes, that is what it means, and yes that is difficult. And it's your job, because anything less is by definition holding the company back.

If you can't handle that, don't be a manager, don't be a leader, and most of all, don't be the CEO. The company—and all its employees—deserve a real leader.

So convert the selfishness and egocentrism of starting a company, needed initially to get the engine turning over, into an egoless, outward facing, empowering, team-growing organization, where your goal is for you to never be the most knowledgeable and experienced person in the room, because you've surrounded yourself with greatness, who each do the same.
source: book

Chapter 79:
Disentangling the three languages: Customers, Product, Business

CONTEXT · ONE TEAM? · THREE LANGUAGES
TRANSLATION

"OK, today we're going to start with some league-building exercises."

credit 976

USING THE RIGHT LANGUAGE FOR EACH CONTEXT

Punchline first:

	CUSTOMER	**PRODUCT**	**BUSINESS**
Language	*Industry-specific jargon in the first person* "I can only use this if..." "I'd get a raise if..." "It would be cool if..." "Everyone needs..."	*Gathering input, deciding, building, deploying* Mission, ICP, positioning, features, releases, workflows, adoption, usage, deprecation, integration	*Financial success* Growth, profit, negative net churn, scale, competition, intellectual property, margins, multiples, enterprise value
Who	Named Customer Personas	PM, PgM, UX, Engineering	Executive team, Finance team, Board
Work	creating, building, designing, troubleshooting, calculating, reporting, organizing	epics, stories, sprints, designs, research, planning, strategy, releases, launches, stakeholder alignment, customer feedback	financial forecasting, budget, hiring plans, project ROI, capital allocation, public company comps

	CUSTOMER	PRODUCT	BUSINESS
Metrics	activation, activity, abandonment, support tickets, time-to-first-success, WAU/MAU, value-creation, reviews	conversion rate, volume, feature-use, API calls/day, deploys/week, sprint velocity, bug backlog, latency, SLOs	MRR (new, canceled, upgrade, downgrade), R40, NRR, ARPU, GPM, LTV, CAC, EV, IRR, Magic Number, variable costs, fixed costs
Translation	**CUSTOMER → PRODUCT** JTBDs and user-stories connect the work of the Customer to the work of Product		**PRODUCT → BUSINESS** Correct strategy + excellent execution → business success

ONE TEAM, YET CROSS-PURPOSES

The CEO asks "Why isn't 'grow revenue' your main goal for the quarter?" She's not wrong; revenue is the reason we're all here. The Product Manager says "My goal is delivering value to the customer, as quickly as possible. Revenue reflects more than that—sales effectiveness, marketing effectiveness, and market swings from things like COVID or inflation or economic pull-back." He's not wrong: "Revenue" is a shared goal and might not be correlated with product development or even customer satisfaction.*

The customer says "of course *everyone* needs [feature], so it's *crazy* that you don't have it." The Product Manager tries to understand (p. 239) the underlying pain, the core "job that is being done." The

* My solution to this "who is responsible for what number" problem is in this article about Product KPIs (p. 645).

customer feels that they've already explained what they want. Six months later, when the feature still isn't added, the customer thinks no one listened. The Product Manager knows that the feature wouldn't make sense to most other customers, but a different update later in the year will solve the use-case in question.

The marketing department asks what features will be delivered by September 17th, because that's when the big global event is happening. The finance team wants projections for the next twelve months, not because they're filling in spreadsheets (although that too), but because projected revenue informs hiring plans in sales and support, and investment plans in marketing, and with a nine-month lag-time for both hiring and training, we have to start that process today (p. 773). The product team has only planned out the next few sprints, because who knows what will happen later in the year; we're *agile!*

THREE LANGUAGES

The key to resolving these (apparent, not real (p. 589)) conflicts is to recognize that there are three distinct "languages" being spoken simultaneously.

I mean "languages" literally—people using different words, even when they're talking about similar things, like the marketing calendar, the fiscal quarters, the sprint planning, the feature-prioritization, and the jobs-to-be-done-that-the-customers-are-doing. And beyond words: the concepts, the goals, the requirements, the specific challenges they face each day.

Conflicts happen when one party applies their language to a party that is using a different language. It doesn't translate automatically. So it's up to us to create systems that not only translate, but that help achieve each others' goals.

Language of the Customer
This is what I'm trying to accomplish. This is what's preventing me from doing my job. This would make me a hero in the eyes of my boss (p. 737) or client. This would make me more money (p. 165). This would increase my reputation. This fulfills a higher-level need (p. 259). This de-risks my project. This is what I will say in a case study. This is what I'm going to say on Twitter. This is what I have written in an online review.

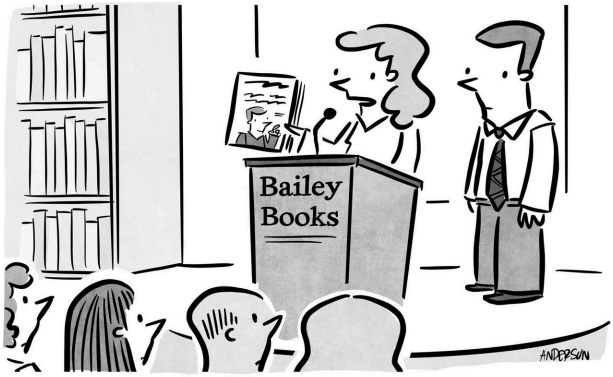
Language of the Product
Strategy: How we will win (p. 489). Competitive analysis and market analysis (p. 71). Long-term versus short-term competitive advantages. Leveraging our strengths (p. 543). Defining the ideal customer (p. 317). Scrum and kanban. Sprints and frequent delivery. Epics and stories and cards and tickets. Tasks and bugs and features. Delivering "customer value" (whatever that means). Retros. Sequencing. Reducing costs. Feature-usage and abandonment.

Language of the Business
Growth. Profitable / sustainable / maximized growth. Profitability. Sales and marketing efficiency. Product ROI. Predictable results. Annual and quarterly planning (p. 1065). Hiring plan. Budget-to-actual. Ratios that increase the stock price relative to current revenue.

TRANSLATION

Of course, everyone is "correct," in the sense that all of this is critical for the success of the business, and they are using appropriate language for their corner of the universe.

Therefore, we need translations to bridge the languages. And often in a high-tech company, it's Product's job (p. 817) to do that.



credit[977]

"You may know our author from his previous best-seller, 'Leveraging Transformative Disruption at the Intersection of Robust Scalable Innovation and Next-level Blue-sky Deliverables: Unpacking Hyper-local Pain Points for the Customer-centric Vision Ecosystem.' But today he'll be discussing his latest book 'Buzzwords, Jargon and Lingo: Confuse your way to the top!'"

Translating from "Customer" to "Product"

The bridge between customer language and product language, in modern terms, is the JTBD (Job To Be Done), further broken down into the "story."

Customers can completely describe their own pain or requests. They often have ideas for features but lack the context and expertise to integrate their desires (p. 711) with the constraints of development, in capacity and against architectural evolution and other customers' desires. Product work requires definition—requirements, requests, and enough of a description that everyone can decide what work might result in the desired outcome.

The high-level translation of the customer's life into product language is the JTBD, in which you project their world onto internal language of your company. For example, in WP Engine's case,* custom-

* Top ten public website platform, as measured by the percent of the highest-traffic ten million domains.

ers who are worried about "hackers," or scared that getting hacked will hit the news or will cause expensive damage (their language), would be translated into JTBD concepts like:

- **Context:** Marketers are scared of security breaches that (a) hurt their brand, (b) cost them time and money.
- **Inciting Event:** After getting hacked once, marketers become highly motivated to move their website to a secure platform, even at increased cost.
- **Job-to-be-done**: My website is fast, scalable, and secure.

While that characterizes the customer's life in company- and product-centric terms, it's not yet a list of things to actually *do*, which is another role of product. Here we use the language of work. A common tool is the "user story," which often comes in forms such as:

- **As a** [who]**, I want** [feature] **so that I** [benefit] **because** [why]**.**

  Applying to the security sample, we might have:

- **As a** [non-technical marketer]**, I want** [a report about how many nefarious attacks were thwarted in the past week] **so that I** [can show my boss] **because** [it proves I'm proactive and intelligent about security; if we get still hacked, my job is still secure]**.**

  Now we can do the work.

(And we can write the sales and marketing copy; this is exactly the bones needed for a sales person to explain both what the product does, and why it matters, from the point of view of the customer.)

Translating from "Product" to "Business"

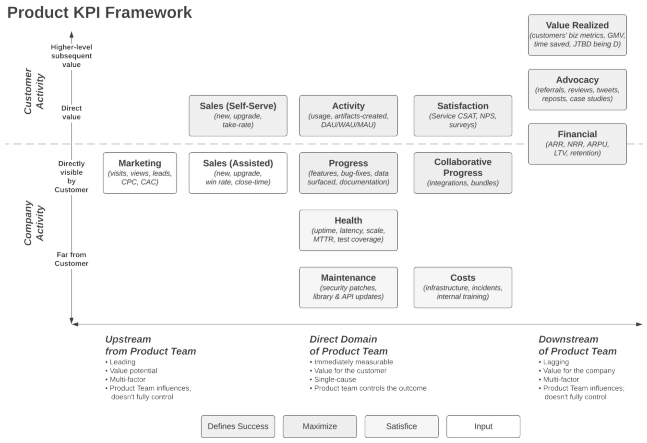
The bridge between product language and business language, shows how the activities of product advance the strategy and goals of the business.

Often the most critical thing the business wants to know is "how much will we grow?" Of course much of this comes from Product:

- Are we positioned intelligently (p. 165) and competitively (p. 259)?
- Are we targeting the right market segment (p. 317)?
- Is our product paying off the promise (p. 949) we made on the website?
- Are we building the right features (p. 239)?
- Do we have the right pricing (p. 515)?
- Are customers delighted by our product (p. 275), so much that they stay forever, growth with us, and advocate for us with word-of-mouth and great reviews?
- Are we expanding into adjacent segments and products (p. 793), at the right time?

But much doesn't:

- Are our marketing campaigns effective?
- Is our sales pitch and sales organization effective?
- Is our white-glove on-boarding team being effective?
- Is our support team increasing retention?
- Is our account management team building high quality relationships?
- Is our corporate development team building high quality partnerships?
- Are we expanding into new geographies?
- Are we budgeted properly across the company?



See the article referenced above for details.

Therefore, it's Product's job to clearly articulate how they are contributing, without having full responsibility for the total impact on the business's goals.

A good way to do this is with this metrics framework (p. 645), in which everyone's KPIs are represented as being important, but with context and areas of responsibility and accountability.

The first step in resolving these apparent conflicts is to recognize that everyone is doing what they think is best, operating from their own area of excellence. The second step is to translate.

Using the right language for the right things creates clarity and insight and helps us work better together.

Chapter 80:
For probabilities, use Fermi numbers, not words

PUZZLE · SOLUTION

PROBABILITY WORDS

Words have fuzzy meanings (Figure 1), but that alone doesn't mean words are useless.

However, we find serious problems when we examine these words in more detail. Another study gives us even more insight (Figure 2).

Note that the median intended probability of the word "likely" is 70%, whereas the intention of "unlikely" is 20%. Since that is the same word, just negated, we should expect the numbers to be symmetric like 70/30 or 80/20, but instead we're biased.*

---

* There is extensive literature on our innate bias regarding both rare and common events; people are surprised how often 5% things happen, and how often 95% don't. Nassim Taleb's *Black Swan* expounds on this, and it's obvious in our everyday experience, like how we get upset when a "10% chance of rain" rains out our picnic.

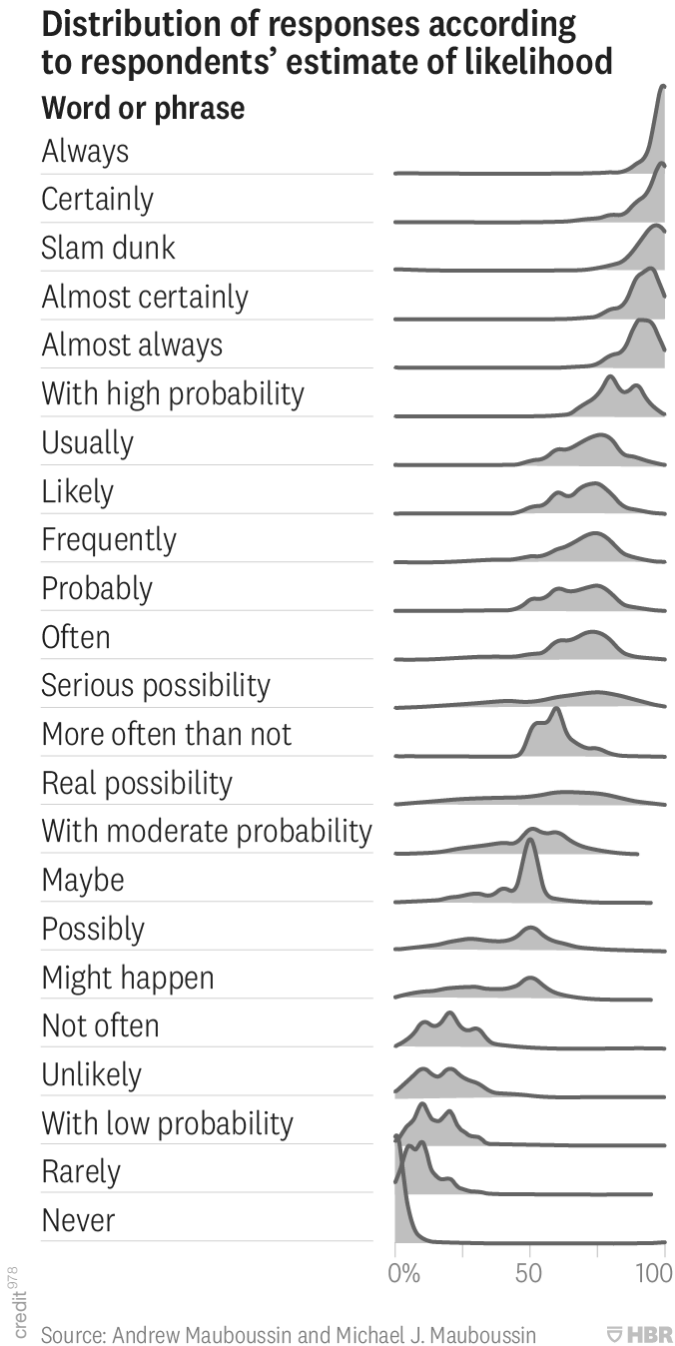


Figure 1: Study: Distribution of numeric probabilities implied by probability-words

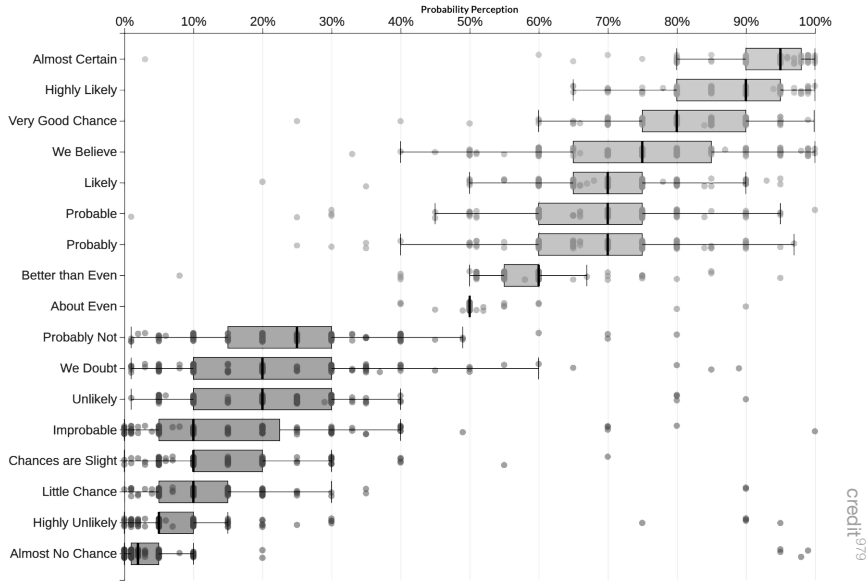


Figure 2

Even worse with "probable / improbable" at 70/10, and strangely while "probable" and "probably" are the same at 70%, "probably not" is 25% while "improbable" is 10%. Perhaps that makes sense to a grammatician, but surely this causes confusion in normal people, especially when we're applying these words in analytical contexts like risk-analysis or debating a strategy.

The communication problem is even worse because *individual people* disagree to an even larger extent. For example, these dots are a single person's evaluation for these words (Figure 3).

Note how pessimistic they are on the positive words, scoring "probable," "probably," and "likely" as only 50%. Then matching the median of 20% with their negatives ("probably not," "we doubt," "unlikely"), but then they claim "improbable" is a straight-up 0% while "highly unlikely" is still 30%.

I feel for this person. Although I admit the formal definition of "improbable" cannot be exactly 0%, what do I think in real life? If the

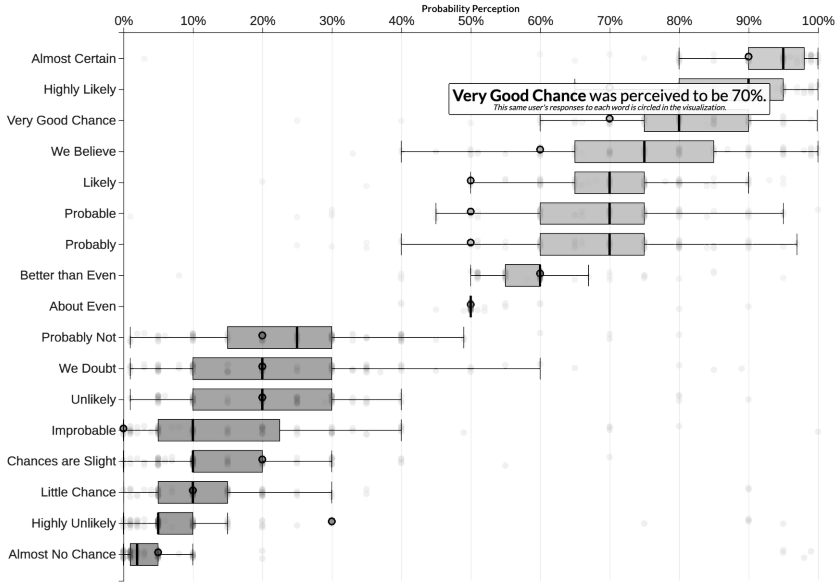


Figure 3

consensus is that some project is "improbable" to complete on-time, I would certainly *act* as though that probability were 0%.

So that's the point: Words don't work, because *the differences between individual interpretations are larger than the differences between the words themselves*!

Furthermore, even if you assign an official numeric probability to words, and train competent people to adhere to those definitions, it still doesn't work. The CIA attempted exactly this,[980] publishing what they call the "Sherman Kent Scale." It explains, for example, that "probable" is defined as 75% and "almost certainly not" is defined as 7%. However, when officers were assessed against the scale, fully half the time they failed to stay in-range (the grey bars) (Figure 4).

We should give up on using words to describe probabilities.

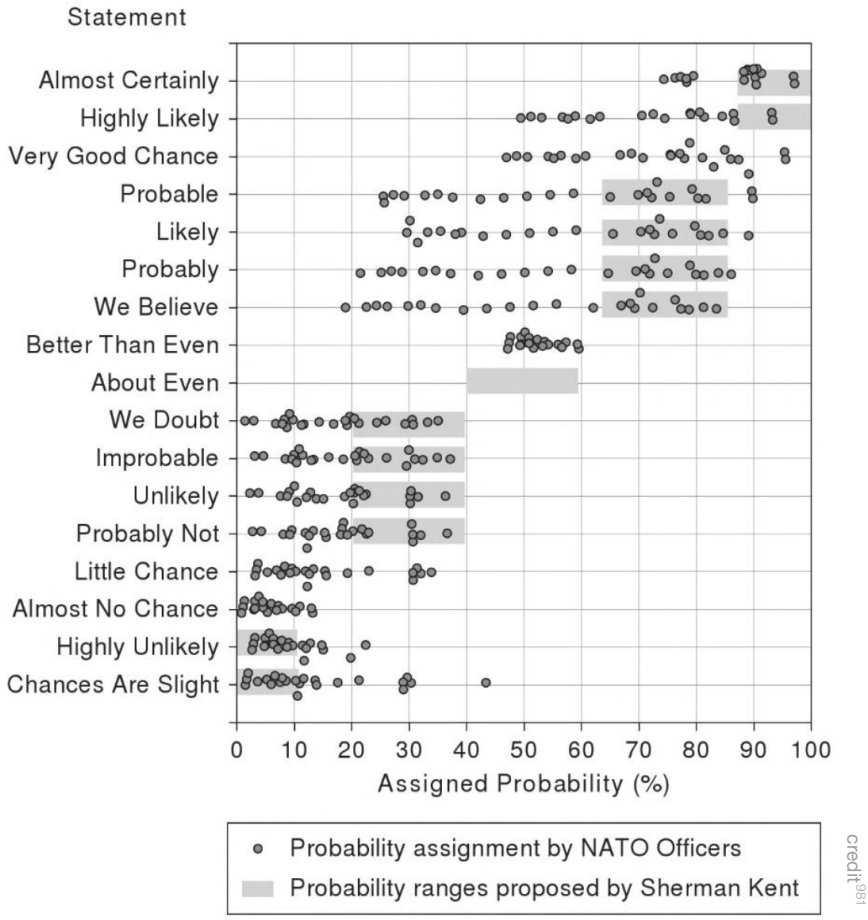


Figure 4: "Clearly, the readers in this experiment were not using the Sherman Kent scale even though they were familiar with it." —*Scott Barclay, author of this study*

THE SOLUTION: SPECIFIC PROBABILITIES, FERMI-STYLE

An obvious solution is to force people to use numeric probabilities, never* using vague words. Indeed, this one rule will already improve communication.

But writing down precise probabilities is hard.

It's hard because usually** we cannot specify the probability with precision. The world is often*** unpredictable (p. 193) even with expertise and data, so we *need* fuzzy ranges of probability to gesture towards our intent.

In this sense, it could appear more accurate to say "the project is unlikely to succeed" exactly because it's unknown whether the true probability is 10% or 40%. Still, given individual interpretations of the word "unlikely," we're not accurately communicating that range.

Furthermore, as we know from Fermi Estimation (p. 171) in domains like "impact" and "time estimation," it's unproductive to haggle over details like "is it a 20% or 30% probability." None of us likely**** knows the true number, and anyway we need crisp signals to make smart decisions.

Therefore, the solution is a Fermi-style probability:

1. Use numbers, not words.
2. Select from a small set of options.

This is the same conclusion the CIA came to in the above-referenced 1964 study. Their mistake was to continue using words, but their categorical probabilities were Fermi:

---

* Oops, I mean 0.2% of the time.

** Oops, I mean 83.6% of the time.

*** Oops, I mean 62.9% of the time.

**** Oops, I mean 9.1% of the time.

Probability Word	Probability Value	Expected Range
Certain	100%	100%
Almost Certain	93%	87% … 99%
Probable	75%	63% … 87%
Chances About Even	50%	40% … 60%
Probably Not	30%	20% … 40%
Almost Certainly Not	7%	2% … 12%
Impossible	0%	0%

My recommendation is to use just a few raw numbers, without words, with instructions for "rounding off" that depend on the context.

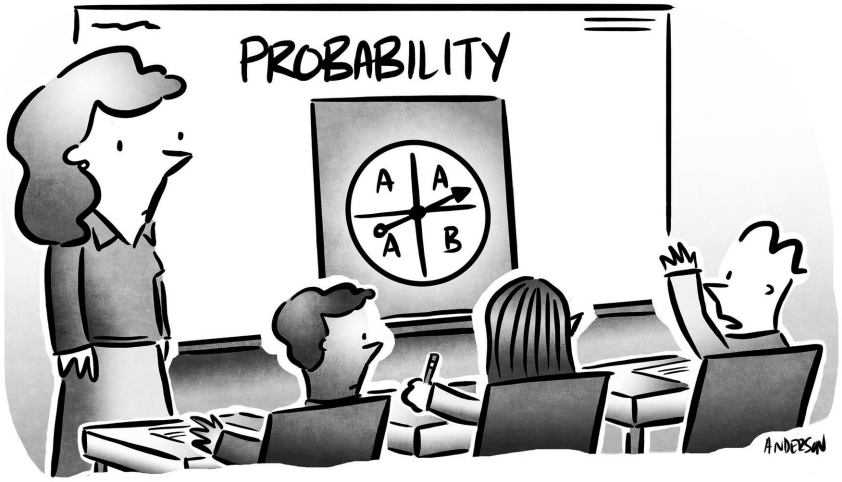
For example, in estimating the likelihood that a project completes on time, we know that in general things are more likely* to be late than early, therefore we should "round off" towards the lower probability.

So go to your "risk" slides, and use Fermi probabilities to force yourself to decide what you think the risk *really* is, so everyone can decide whether or not to act.

Go to your strategy, and put Fermi probabilities on each assertion, so people know what is more or less likely to change as we learn and grow.

Go to your work-prioritization system, and put Fermi probabilities on the "value" or "estimate" metrics that you're using for decision-support. Ask everyone in the team to supply their own numbers independently; where there's disagreement, that's worth a discussion; where there's agreement, you can save time by just moving along. Or maybe don't, because you realize that probability isn't even applicable to such things (p. 1141).

* Oops, I mean more than 51% of the time.



credit 982

"I know mathematically that A is more likely, but I gotta say, I feel like B wants it more."

Perhaps the real reason we use wishy-washy probability words is because we really don't know the probability, and rather than admitting that, we just glide past the challenge. That's the worst reason of all.

Now you have no excuse.

Be brave, and put a number on it.

Chapter 81:
Startup identity & the sadness of a successful exit

PEAKED · IN GOOD COMPANY · IDENTITY

"No, it's fine, I've just never seen coupons used in an acquisition."

credit 983

AT THE PEAK, THE ONLY PLACE TO GO IS DOWN

My fingers trembled as I fed page 34 of 72 into the fax machine, deftly pressing the head of each page into its creaky jaws one at a time so that this shitty cheap machine wouldn't snag two pages at once, slantways, obscuring the precious scribblings adorning the footer of each page where it read: "Seller's Initials: _____".

This is what the last six years were all for. All the labor. All the risk. The brave face for the troops. The self-inflicted unflagging optimism despite little supporting evidence. All those sleepless nights worried about making payroll with a one-time revenue business model where every month you have to find new revenue from new customers. The 10,000 deliberate hours of becoming an expert (p. 1529) in something. The experience you get just after you need it. The hard lessons you occasionally glean from experience. The inner doubt suppressed for the morale of the team. No salary followed by low salary. The "eat what we kill" mentality. The scrounging and scrabbling and begging and fighting the assholes (p. 737) for those morsels of revenue, those crumbs of validation.

It's over. We did it. I did it. American dream? Check.

The 73rd page spat out confirming the successful transfer of the previous 72.

And then... *sadness*.

A profound sadness. Not depression—not hopeless or rudderless—but pure sadness, when your lungs sink into your belly, the punch-in-the-stomach of discovering your dog was hit by a car or that your dad is terminally ill.

"What the fuck?" I thought, "Why am I feeling this? I'm supposed to be feel... happy? I guess? Something other than this."

IN GOOD COMPANY

Almost all startup founders experience a deep and prolonged sadness after selling their company, even when the sale is an outrageous success. Why?

The answer is important and fundamental for all startup founders, whether or not they intend to sell their company some day.

It is more than generic malaise. It is a fundamental disconnection from what it is to be a human being, to be yourself, to drive towards something. Combined with the guilt of success, or at least the knowledge that you can never complain about not knowing what to do with the kind of time and money that nearly everyone else on Earth can only dream of.

And so you have mental break-downs like Vinay Hiremath, who sold his company Loom for nearly a billion dollars, nearly dying in the Himalayas[984] and driving away his girlfriend (Figure 1); Markus "Notch" Persson, creator of Minecraft, who melted down on Twitter after selling to Microsoft[985] for more than two billion dollars, after he found himself directionless and socially distant from everyone he knew, from employees to friends to girlfriends (Figure 2). There are many other stories[986] from less famous[987] but no less valid entrepreneurs.

I am rich and have no idea what to do with my life

Life has been a haze this last year. After selling my company, I find myself in the totally un-relatable position of never having to work again. Everything feels like a side quest, but not in an inspiring way. I don't have the same base desires driving me to make money or gain status. I have infinite freedom, yet I don't know what to do with it, and, honestly, I'm not the most optimistic about life.

Figure 1: Loom founder on his blog,[988] after selling for $975,000,000.

And of course it's not just entrepreneurs—it's anyone who devotes their life and identity to a cause which then suddenly vanishes. The quintessential and well-studied case are Olympic athletes, where a single-minded drive to be the best in the world, starting young and therefore occupying most of their conscious lives, reaches a pinnacle in whatever Olympic games is their last, whether they've won gold or not. The individual stories come both from the most decorated athletes in their sport* and from everyone else who is objectively world-class.** Many studies*** show that these anecdotes are examples of the rule, not exceptions.

Commiseration helps us understand that it's normal, but it doesn't help us understand why it's happening, and therefore what to do next. For that, we have to put our finger on what's going on.

---

* e.g. Michael Phelps, especially in the HBO documentary "Weight of Gold" (2020); Lindsey Vonn in her autobiography *Rise: My Story* (2022).

** e.g. Allison Schmitt in swimming, McKayla Maroney in gymnastics, Gracie Gold in figure skating, all vocal about their struggles as part of their new journey as mental health advocates.

*** The British Journal of Sports Medicine, "Mental health in elite athletes: International Olympic Committee consensus statement" (2019).[990] The IOC's own consensus statements[991] include the "post-Olympic blues".[992]

Figure 2: Minecraft founder on Twitter,[989] after selling for $2,500,000,000.

IDENTITY

"I sell my artwork on Etsy. Want to see?"
–Barista at an Austin coffee shop

If you ask her, "Who are you?" She would answer: "I'm an artist."

If you ask her, "What do you do for a living?" She would answer: "I'm a barista, but that's just my day job. Want to see my artwork?"

Is she a barista because she pours coffee for money? Is she a driver because she drives a car to work? Is she a maid because she cleans her own apartment?

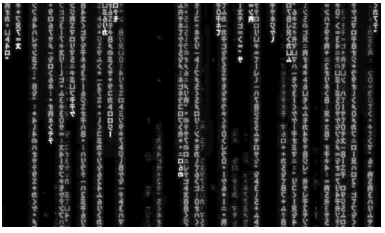
No, **she's an artist** because **that's what she *really* is**. "Barista" is one of many necessary means to the ends, where the "ends" are the basic human needs, followed by creating art, I guess with some Maslow layers (p. 259) in-between.

A startup founder lacks this distinction between personal identity and work identity, and this is the key to the "sale-blues" phenomenon and other behavior.

A startup *is* the founder's personal identity. Startups are not something you do to make ends meet or a "necessary evil" *en route* to what you "really" want to do.

A startup is an obsession. You do it because you couldn't stop yourself. Because when you were doing anything else, you were thinking about it. *That* is the mark of "who you are." Interviewers ask me "Why did you decide to do a second and eventually a fourth startup?" And the answer is "For the same reason that I started the first one—because it's in my DNA and I have to do it."

What do you do in your spare time when you have a startup? What spare time? This is all your time. It's not just the last thing you think about before falling a sleep, it's the thing that won't *let* you sleep. It's the first thing that trickles into your brain in the morning like The Matrix patterns filling the void.



It's why you can weather the painful thoughts like the ones whistling through my ears while I fed legalese into a rickety fax machine. You are consumed, this is your life, this *is* you. There's no room for anything else.

When you sell your company, others are quick to throw jabs like "So, you sold your baby?" Which means: "You're a sell-out (p. 45)." (Or: "I'm jealous." Or both.)

It's less like selling your baby and more like selling your own identity. Now you have to find a new one.

If there is a new one.

And there's no particular reason to think there will be. Which is part of the fear.

Speaking of babies, this all sounds a lot like "baby blues"—depression caused by elevated levels of the enzyme monoamine oxidase A—that 70% of women experience after giving birth. A third of those women will experience this for up to a year (postpartum depression).

It's characterized as a feeling of loss and of mourning, which is seemingly at odds with the arrival of a new life which is just the opposite—gain and celebration. This intellectual dissonance creates secondary emotional effects, specifically the devastating belief that "I must be a bad mom for being sad that I have a new baby."

Are the baby blues the same emotional effect as selling a company? Maybe not—I already don't like saying "a startup is like a baby."[993] Besides, postpartum depression is triggered by the arrival of responsibility and, if you insist on the baby/startup analogy, selling your company is the departure of responsibility.

But one thing that definitely is the same is that "feeling of loss and mourning." **A piece of yourself has been eviscerated, irrevocably.**

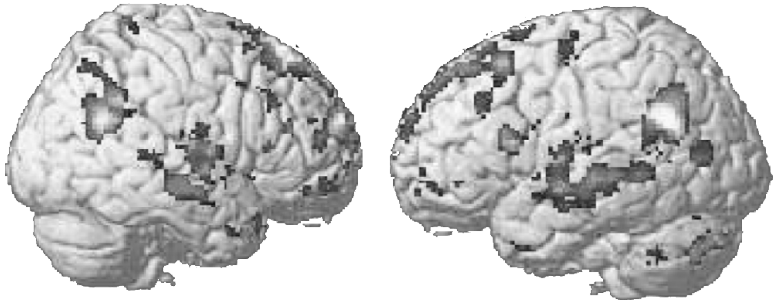
And, in fact, maybe it really is like your baby. In one study,* researchers watched brain activity in people who were both entrepreneurs and parents. Comparing activity across neutral images, those of other children or company logos, or their own children and company logos (Figure 3), they found that the brain activity when exposed to their firm's brands were similar to the activity when gazing at their own children, and of course both different from the other images (Figure 4). In short, our brains really do react to our companies the way they react to our children—the love, the attachment, the intensity, even the threats and risks and worries.

---

* Marja-Liisa Halko, Tom Lahti, Kaisa Hytönen, Iiro P. Jääskeläinen, "Entrepreneurial and parental love—are they the same?" (2017)[994]

credit[995]

Figure 3: Protocol showing images to parent-entrepreneurs — neutral, other children and brands, and their own children and brands.



credit[996]

Figure 4: The areas of the brain that light up when viewing your own children *or* your own brands, different from other images.

As we showed earlier, it's not only a normal feeling, it's by far the majority case. In a high-quality study[997] from Columbia Business School, out of 22 entrepreneurs who sold their companies, *every one*

of them experienced this effect. Some refocused the energy into the Next Thing (almost always new ventures), but most took *years* to find that thing which could replace not only the excitement but identity, and many still haven't found that Next Thing at all. Many wished they could have their companies back; a few did buy them back.

Selling their companies forced them to answer a difficult question: If you could do anything, what would you do? What's *really* important to you, as opposed to a job? What do you *really* want to be when you *really* grow up?

The problem is, **the startup already *was* that thing!** It's a grinding, batshit-crazy, risky, irrational, epic adventure. You wouldn't have done all that in the first place if that wasn't already the thing you wanted to do. A startup is never the easiest career path.

Does this lead to the conclusion that selling a startup is the wrong choice much of the time?

No. And this is perhaps an ever sadder realization.

It's an easier decision if you're saying "I can't wait to start the next thing" but perhaps you're thinking "I don't know what I'll do with myself." You must resist the urge to believe that getting millions of dollars will make you fulfilled, or happy. "Money doesn't make you happy" is cliché because it's true. You need to understand how to be fulfilled (p. 399) regardless.

But all that doesn't mean selling is wrong.

Just like a bad relationship that has to be ended even though it will be painful, especially if you genuinely love the other person: Just because it's painful doesn't mean you don't need to do it.

Building a company in year-one is completely different than building that company through year seven (when I left Smart Bear) or year fourteen (when I left WP Engine). The CEO's job description changes over time, and so does the company, whether you sell it or not. Are you emotionally prepared for this as well?

Are you OK with innovation taking a back seat to developing scalable, mature processes? Are you OK releasing control in day-to-day operations to managers, and then releasing control of the managers

to your executive team? Are you OK wresting yourself out of Visual Studio, out of Adwords, off the website, off the live chat, out of the sales calls, trusting your managers and not being that sort of meddling micro-manager boss that you yourself hate? Are you OK shouldering the burden of the livelihoods of dozens or thousands of families rather than "pulling 90 hour weeks" to push some code out the door with a co-founder, which now seems easy in comparison?

The fact is, successful startups grow up. They grow into businesses and mature into sustainability, risk-avoidance, HR law, strategic planning, executive meetings, and all that. The founder-CEO is still steering the ship but it's a different sort of ship (p. 827).

What to do? I wish the answer weren't "it depends" or "every choice is a path of pain," but often that's what it is.

In my case, it changed over time. At Smart Bear I didn't want to lead a huge company. I didn't want to relinquish Eclipse and my ability to check in code. I didn't want to manage managers or figure out what changes, strategies, hiring, products, marketing, and sales were needed to make $100M/year. So for me I sold at a good point: before I needed a C_O, but after the company was big enough to garner enough money to cross the Freedom Line (p. 45).

Now at WP Engine, I have new ambitions and inclinations. I am now that CEO* who manages managers, who sets vision and direction but not day to day operations, who worries about company culture[998] but who doesn't have SSH access to all the servers, and who is driving towards a company with products and a market and a team which we believe can indeed generate $100M/year.**

That's exciting to me. This is my new challenge. I will always love writing code and getting a company from $0 to $1M/year.

---

* Editor's note: Written in early 2013, later that year the wonderful Heather Brunner joined as our CEO, still leading the company 10 years later, as what I've since called (p. 399) a "late-joining founder."

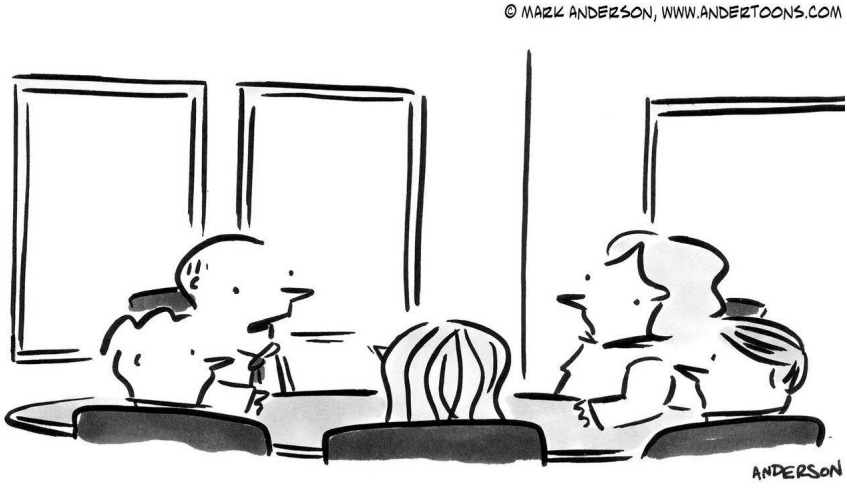
** Editor's note: It did, at hyper-growth velocity. Now, in 2023, our revenue is many times that figure.

But, today, right now, for reasons unknown even to me, *this* is who I am.

> I went to find the pot of gold
> That's waiting where the rainbow ends.
> I searched and searched and searched and searched
> And searched and searched, and then...
> There it was, deep in the grass,
> Under an old and twisty bough.
> It's mine, it's mine, it's mine at last...
> What do I search for now?
> —Shel Silverstein, *Where the Sidewalk Ends*

Editor's note: I would later develop a personal framework *(p. 399)* *for avoiding burn-out and staying with the company for the long haul. That story and system might help you if you're struggling with this now!*

Chapter 82: Productive meeting activities: Leverage the team, empower the individual



"OK, now that we all agree, let's all go back to our desks and discuss why this won't work."

We all hate the meeting where we prove once again that a committee is a beast with twenty legs and no brain, or when we could've just read the materials quietly to ourselves at a more convenient time (and maybe we did, and it's being read to us *again*), but there's also the elation of *espirit de corps* and accomplishment following a productive meeting, where in actual fact "all of us were smarter than any of us."

Even when an enlightening discussion ensues, sometimes the person who requested the discussion gets foiled. A person wants to solicit ideas, but ends up getting told which idea is best. A person wants help deciding between two vetted options, but instead receives seven new options.

How can we leverage the wisdom of the crowd (p. 931), while ensuring that decisions and other responsibilities continue to reside with an individual?

Some answers follow. Enjoy!

Ranting

I need to get something off my chest, in a trusted environment, where I can say things that make me vulnerable (e.g. "I don't know where to start" / "I'm scared to face this"), or might sound frightening to others (e.g. "I don't think what that other team is doing is right" / "I'm doubting my strategy and I'm worried how my team will react if I broach it").

I don't want you fix the problem, I just want you to understand the problem. I wouldn't mind if you supported me while I work through the problem. For example, I have "people problems" in my organization; I don't need everyone to jump in, but I do need them to "be kind" to my teams while we work through this, and understand that we might need extra time to do our work.

This can lift a great emotional weight off the person, while increasing team cohesion, trust, vulnerability, and empathy, therefore it is a fantastic use of meeting time.

Ideation

I need ideas. Possible causes of a problem, possible solutions of a puzzle, more options for actions I could take. I'm stuck right now, too close to the problem, and I don't have ideas, or I'm tired of looking at my ideas and I need new ones. I sent an email about this ahead of time, so the group has had time to think about it.

I don't want you to solve the puzzle or decide which option is best, I just want you to help me see options that I'm not currently seeing. For example, I have a strategic challenge where I need to solve for two apparently-contradictory things, and I need ideas for how that might work. I don't want you to decide my strategy for me, but help me see more possibilities.

Everyone gets stuck; no one can have all the great ideas by themselves. Having a space where ideas can flow but no one is trying to tell you want to do, is fun for everyone and impossible to do alone, therefore it is a fantastic use of meeting time.

Decision-Support

I need help deciding something. I know what my options are, and I've analyzed them (sent ahead to the group), but it's complex and I'm uncertain which to pick. I might even be uncertain *how* to decide. It could be that the best option isn't yet in the list, so it could be good if we synthesize something new. But ultimately I don't want seven new ideas; I want to make a decision.

I don't want you to tell me what to pick, but I do want input, whether on the evaluation of the options, or on what basis I should make a final decision. I might even ask the room to vote; not to dictate my future, but because I trust the room's opinion and I want it as another input in my decision process.

No matter how many prioritization articles (p. 221) you read and ROI rubrics (p. 171) you make and multi-dimensional maximization charts (p. 603) you make, complex decisions are difficult. Seeing your choices re-

flected in others' experiences and points of view can help, and cannot be done alone, therefore it is a fantastic use of meeting time.

Sparring Practice

I need a Devil's Advocate to sharpen my work. I want find the weaknesses in my strategy, my feature concept, my keynote presentation, my reorg proposal, my new UX design; I need a safe space where people "attack" my work, to surface legitimate flaws, or to reveal ways of making my argument that deflect potential attacks before they begin.

I don't want you do my work for me. If you have an idea for improvement I'd love to hear it, but I will decide what to do with that idea. It's OK if you "attack" from a perspective that isn't even accurate, because my future readers might have the same misunderstanding, and I want to be clear even to them. I want to be challenged with the intention of making my work bullet-proof.

Whenever you see a brief, tight, bullet-proof document, it took a lot of work to get there. No one by themselves can think of all the ways that others will (mis)interpret ideas, or think of all the legitimately good counter-arguments that deserve answering, therefore it is a fantastic use of meeting time.

Declare the conversation

It's counterproductive when someone wants a Ranting, but receives a Sparring Practice. Declare what kind of conversation you want, then everyone can participate in the right way, and facilitators can reroute the conversation when it inevitably strays from the parameters of the discussion.

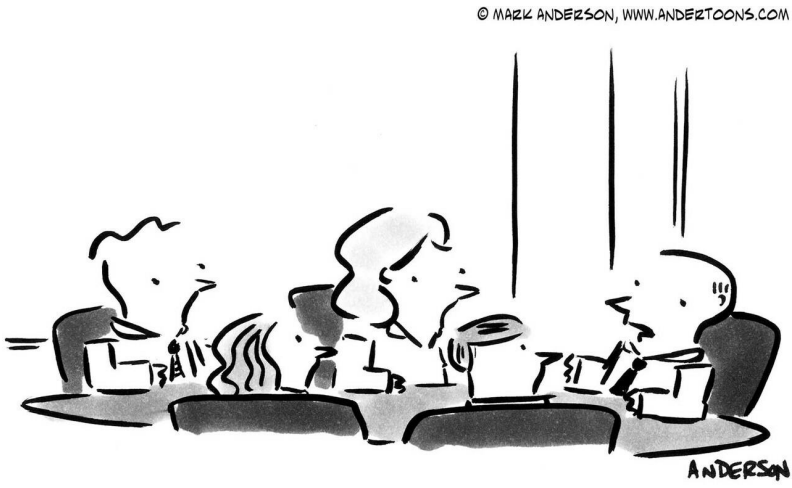
There is an "I" in Team

Notice that everything here is selfish—each exercise is about what one person needs.

This is contrary to how we often work together. We say things like "We need to decide on our Q2 priorities," instead of "Person X will decide the Q2 priorities by Friday, so let's discuss to ensure everyone is heard, and the best ideas are incorporated."

It's healthy for X to seek a consensus, and X often genuinely wants to be influenced by teammates who X trusts and respects, particularly when opinions are unanimous. However, a decision made by "us" is owned by "no one," when in actual fact it *is* owned by X. Acknowledging this reality is useful, even as you also—correctly!—spend 98% of the conversation discussing as equals.

This is also a healthy pressure against top-down control, in cases where the decision is supposed to reside in a self-managed and empowered team. That is, suppose person P is responsible for some project, and asks for help in a room containing peers and person Q, who is P's boss. The right way for the entire room to help P are techniques like the above, where P leverages the brainpower of the room, but stays in command (p. 413), keeps the responsibility, keeps the autonomy, even with Q present. It is a wonderful experience for a leader to assist without usurping.



credit[1001]

"It seems I've run out of sports analogies, so meeting adjourned."

Let's put more ideas for constructive meeting activities in a Twitter thread![1000]

Chapter 83: Our unhealthy fixation with emulating #1

Many sporting contests culminate in a single-elimination tournament, where each match produces a winner who continues on, and a loser who goes home (Figure 1).

At the end, only one contender will be undefeated, earning them the crown of "undisputed champion".

But this neither as clear nor as fair as it seems (as any sports fan will agree, as they start rattling off examples). **Being "undefeated" is merely a result of the system** itself, not necessarily proof that the winner is in fact better than other top contenders.

To see why, consider this extreme variant: The "sporting contest" is actually just a coin flip. The tournament will work the same, with half the teams losing in the first round, half of those remaining exit in the second, and so forth, until the final match where one team will be undefeated.

Of course in this example the team earning the title of "number one" is random because the *contest* is random, but the distribution of how many teams made it to each stage and the fact that exactly one

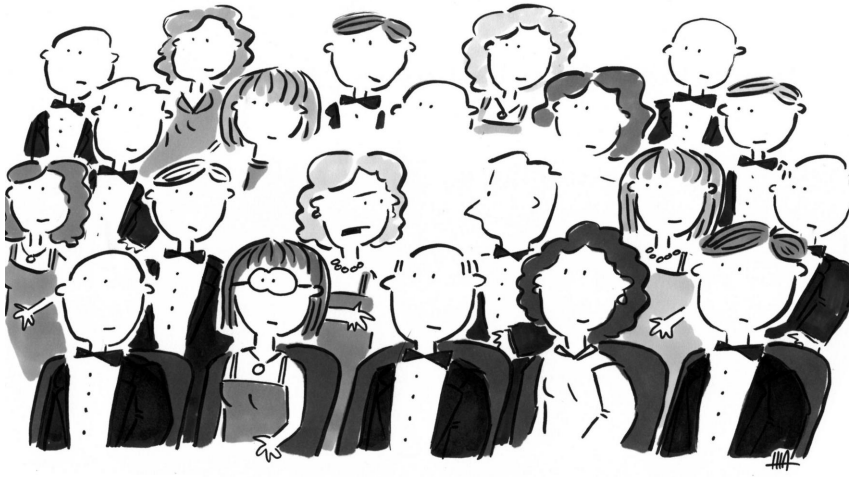
Figure 1

went undefeated is dictated by the *tournament system*, regardless of how random each *contest* is.

The outcome of most sporting events are a combination of skill and luck in unknowable proportions. This is why sports are exciting and suspenseful; on any given day a lesser-skilled underdog can win (whether by might or by luck). You could mitigate the luck factor by replaying a match a number of times and declare win/lose/tie based on statistical significance at a predetermined confidence level, but of course this is neither fun nor practical.

Business is also a combination of skill and luck in unknowable proportions (p. 1035), and it's also clear that some tournament system of markets and economics determines the distribution of results, and that it's an experiment we can run only once. The very fact that long tail distributions are common in many industries indicates that there are *systematic* effects outside the typical "luck versus skill" debate.

We tend to fixate on whoever is #1, in business as with sports, tacitly assuming that the contest is mostly skill and therefore the tournament has selected the rightful leader. But I'm not so sure we know the



credit[1002]

"No one's winning. It's ballet."

skill/luck proportion. I'm not sure we can assume the contest (marketing, sales, product) and tournament (the marketplace) picks #1 based on a repeatable, codify-able law. Same with #2 or anyone else.

Does the winner of a pitch contest have a better shot at their business than #7 who learned from the experience and is now redoubling his efforts?

Does McDonald's have all the answers or does the #12 largest fast-food chain in the world also have something to teach us?

Isn't the #4253 largest company in the world still a success story, perhaps with new lessons and perspectives?

If we reset the clock and the Google guys published their PageRank paper at MIT instead of Stanford, and didn't get plugged into the Silicon Valley system, would Google have existed?

How do we know which decisions were important, that actually caused success?

Alright, so what? **When does this matter?**

credit[1003]

"I'm sorry, Jim, I'm afraid this is only a win win situation.

When someone insists you need to be "more like Google," consider that perhaps it's the only thing they know to compare to.

When someone insists you need to be "more like 37signals," consider that almost no successful companies are like 37signals.

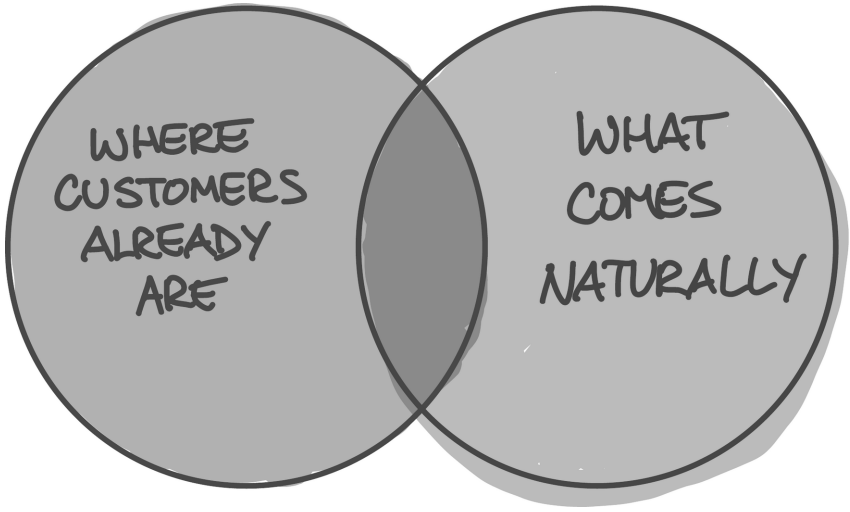
When someone insists you need to be "more like Apple," consider that they probably have no idea what really goes on inside Apple, or whether they're anything like you. Also, do they mean "Apple, today" or "Apple when they were 3 years old, like you, and doing hardware with the mindset of the late 1970s"?

No. More interesting is when someone suggests that you remind them of this other little company you've never heard of, but when you visit their website and try their product you realize it's resonating with you, that this feels like a finer, more mature version of yourself, that you're getting reinvigorated about your own business not because of their top-line revenue or celebrity status but because they're inspiring you to become a better version (p. 751) of what you already are.

It's fine to muse about being #1, but let's not all strive to become just like #1.

Chapter 84:
How to select your first marketing channel

THEM · YOU · BUT WHAT IF…



credit 1004

Figure 1: Scott Brinker's Marketing Technology Landscape from 2018; that was the last year it was "small" enough to be displayed in this format.

You've built a product, and it even works! Mostly! Now, how do you select your first marketing channel?

There are dozens to pick from, and hundreds of products and services to help you execute each one (Figure 1).

But there *aren't* dozens of marketing channels that make sense *for you, right now*. At the beginning you are beset by constraints of time, money, brand, reach, and specific knowledge about your customers (p. 239). These constraints eliminate most channels as non-viable.

The bad news is: You have fewer options than you'd like. The good news is: You can leverage those same constraints to answer the question: Which marketing channel is right for me?

There are two primary constraints relevant to this question:

1. Where your customers already are.
2. What comes naturally to you.

credit 1005

"I miss the old days when we could just tout things."

1. WHERE YOUR CUSTOMERS ALREADY ARE

You don't control where your customers already go for industry news or to discover new software. You have to fish where the fish are.

For example, if you're selling accounting software, most of your customers will be on LinkedIn, some on Twitter, occasionally on Threads, maybe Facebook if it's for, let us say, a more mature connoisseur. But not Pinterest, not Snapchat, and not TikTok. Besides social channels, which keywords do they search for when they encounter the problem that you solve? Which podcasts do they listen to? Which YouTube channels do they subscribe to? Which consultants do

they already pay to solve problems like these? (The answers to some of these will be "none".)

On the other hand, if you're selling all-natural fair-trade miniature-alpaca yarn to weavers, they might be on Facebook, Pinterest, and possibly Threads.* There might be weaving clubs that would be better targets than selling directly to consumers. There might be magazines (even in print!) and certainly some popular influencers.

Because you don't control this, it is a constraint. You have to go where customers are already engaged.

2. WHAT COMES NATURALLY TO YOU

This sounds self-indulgent, contrary to typical business advice. Aren't you supposed to get out of your comfort zone, learn new skills, dive into new media, and become an "expert" (p. 1529) in three months through a crucible of a hundred intense hours of coursework and enabling software?

* I don't make the rules, I only make the puns.

No. Doing something unnatural, that you despise, will fail. Think about the social network you do like. You instantly notice imposters—people who don't understand the medium, ham-handedly exploiting it for selfish marketing. It's awkward, it gets no engagement, and in fact you might not even see it as the algorithm silently buries their posts.

Plus, you're going to hate using a social media platform that you disdain. Taking myself as an example: I don't understand Pinterest. I don't understand collecting pictures on boards, I don't understand what happens with boards, I don't understand how you share things, I'm not even sure it's called a "board" and I don't care what it's actually called. If I tried to do it, not only would I be bad at it, I would hate doing it. My disdain would leak onto the page, making the whole effort even less effective. Ineffective and unhappy—is this why you started a company?

So, take stock of your Pivot Points (p. 569), and consider only those marketing options that you're naturally good at, or genuinely wish to become great at; you will select your first marketing channel from that subset.

BUT WHAT IF…

So we've come to the obvious-in-hindsight conclusion that you should operate at the intersection of where the customers already are, and what comes to you naturally and joyfully. With practice, you'll move

from being good to being great, which was unlikely to happen had you selected a channel that wasn't a good fit.

This is smart even if your chosen channel is low-volume. Suppose there's a small-but-vocal community on Threads, but you know that most of your customers are on LinkedIn. But you don't understand LinkedIn—it all looks like AI to you, you don't understand why anyone would read it, it all seems fake, you just can't get your head around it. It is better to be one of the top three creators on this topic on Threads than to be the 200th biggest creator on LinkedIn, even though LinkedIn has many more people. Algorithms don't like promoting the 200th-biggest thing; they do like promoting the 3rd-biggest thing. And you can be 10x or 100x more effective in your messaging. So even though you're talking to a smaller audience, you can be successful.

And then, once you become powerful on one medium,[1006] you can expand to others (p. 793). With a foothold on Threads, and revenue that can pay for consulting or tools or hiring a part-time marketer, you can repurpose your content from Threads but in the style of LinkedIn. Now you're expanding into a second marketing channel from a position of strength.

Another possibility is that the Venn diagram intersection is empty—your customers are only on Facebook and you absolutely despise Facebook with every fiber of your being. What do you do?

One conclusion is that this is the wrong business for you. You just said that you're so different from your customers (p. 533) that you can't relate to them. Perhaps that means you're not the right person to run this business. The business might be great, just for somebody else, not for you.

Another possibility is you need to partner with someone for whom this particular marketing channel is their superpower. Someone who's an absolute wizard at Facebook, accomplishing things everyone else says are impossible. What takes other people a year, they complete in a month. What takes other people two months, they did last weekend. Since you've stipulated this is the most important channel, and that you're going to be terrible at it, consider how powerful it would be if you focused on the areas of the business that are your superpowers, while this other person hauls in new customers every day. This could be exactly the team that maximizes the chance that this company is really going to work.

In any case, you want to pick one marketing channel and go deep.[1007] Not spread yourself over several channels, none of which work well.

Notice that in all these cases, the answer isn't for you to force yourself to figure out Facebook. Of course you *could*. This is your life, this is your company, this is your idea; you can do whatever you want.

You're just adding even more risk to a venture that's already very risky (p. 667), and that's not the right way to solve the equation.

Chapter 85:
How much of success is luck?

"Serendipity is up, fluke is doing well, but I'm a little concerned about our dumb luck."

It's amazing how often "luck" comes up when people find out I started Smart Bear.

> "You're lucky to have your own business. I hate my boss."
>
> "You're lucky your business is still doing okay in this recession."
>
> "You're lucky that you sold your business (p. 45) when you could."

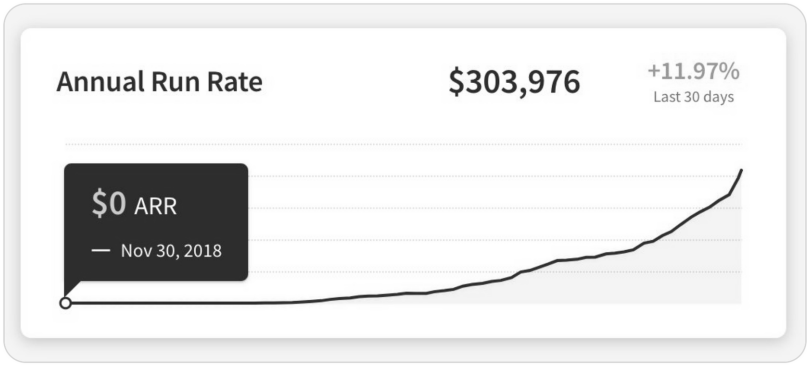
I wanted them to be impressed with the hours I put in, with the ideas I had, with my amazing product-sense, with the way I handled customers, with my 50% sales close-rate, and how I overcame the mental stress of bootstrapping.

But no, success is "lucky." A successful business is a lottery ticket.

It's dismissive, even insulting: It wasn't you, it was luck. Your decisions weren't important, your ideas weren't special, the long hours didn't matter—you're just lucky. Anyone in your place would have done the same; time and chance happeneth to them all.[1009]

It's easy to reflect back that dismissive indignation:

> So when I quit my job and worked 60 hours a week with no pay for years, sacrificing my health (p. 1547), and finally clawed my way out, that was *luck?*
>
> So when I invented a unique product and built it from scratch in a market that didn't even exist yet (p. 71), and people not only needed it (p. 259) but wanted it, and genuinely wanted to support us (p. 275), that was *luck?*
>
> So when I cultivated relationships with the best customers (p. 317) and truly listened to their needs using a specific framework (p. 239), that was *luck?*
>
> So when I had the chance to sell my company (p. 45) at a fair



credit 1010

> price and negotiated a deal that put more money in the pockets of my employees than any other job would have, that was *luck?*

These retorts are fair. But both things can be true. These demonstrate that it's not *only* luck—it is *not* identical to a lottery ticket—but isn't there luck buoying up all aspects of the endeavor?

Yes I cultivated relationships with customers, but wasn't it lucky that the customers showed up in the first place? Yes, they found me through my Google Ads, but wasn't it lucky that I started Smart Bear right when AdWords were new and cheap, when everyone used Google but AdWords weren't saturated with garbage? Yes, I chose effective, brief marketing messages (p. 1439) with an authentic voice (p. 627), but wasn't it lucky that I had a mentor who had already taught me how to do that?*

* Update in 2024: I started WP Engine a year after writing this article; fourteen years after that, it's been a unicorn for years (p. 9), 100x larger than Smart Bear

In fact, I can pick any decision in the history of Smart Bear and the same rhetorical pattern appears: Success tied to something intentional,** but also leveraging something lucky.

Besides this operational luck, there's personal luck, which many people call "privilege" and which Warren Buffett calls "winning the generic lottery"—the luck of being born in this era (not in the Middle Ages), in this country (the US in my case, easily the most supportive place to start a company), to this family (I had a stable household), with certain talents (e.g. building software, problem-solving, being tenacious, having focus), and with this ethnicity, gender, and with all body parts in good working order. While none of this negates the value or necessity of hard work and good choices, neither can you ignore these tailwinds in the final equation of "success."

My conclusion:

Good luck and bad luck are constantly swirling around you. How you *use* it, is not luck.

These individual successes are a result of taking advantage of good luck. What about failures?

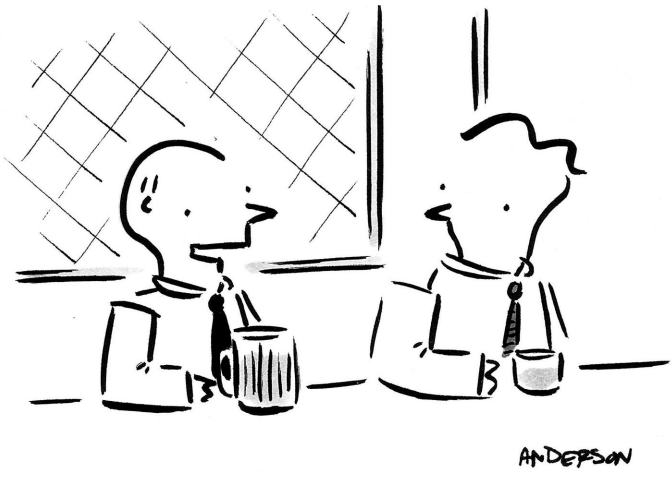
At Smart Bear, lots of marketing and advertising attempts flopped. Ads in certain magazines bombed. (I'm withholding names; print media is having a tough time as it is.***) In some cases I spent many

---

ever was. Once again, it was a great idea, great execution, an immense amount of effort from thousands of people over the years... but also luck: Lucky that the WordPress market in which we operate remained healthy and fast-growing throughout our journey, lucky that people in the WordPress community give us the benefit of the doubt at first and supported us later, lucky that we were able to hire so many incredibly talented people, lucky that it took serious competitors ten years before they really understood how big our market is, and the list continues.

** Just because it's intentional, doesn't mean everything worked on the first try. Indeed, almost nothing works on the first try. You have to experiment and fail (p. 1261) before you find success; pushing through that is tenacity (p. 429), but there's luck in how quickly you find that success, or if you ever do, and if you can even tell the difference (p. 159).

*** Update in 2024: Every single magazine that I used to advertise in, has gone out of business.



"What luck! One of my personalities is a psychiatrist!"

credit 1011

thousands of dollars—which at the time was a significant percentage of revenues—on ads that didn't net a single sale.

Ads that utterly fail in one magazine when they worked in another—that's bad luck. The choice to cancel some ads and not others is not luck. In fact, ensuring that we could measure the efficacy of individual print ads was also a choice. Had we not done this, we wouldn't have been able to distinguish success from failure, and then indeed our destiny would be controlled by luck alone. There are lots of ways to "fail" without failure (p. 1261).

Overall success in business doesn't only mean you "got lucky," it means you *used* luck, taking advantage of the good, identifying and ejecting the bad.

"Luck" rarely comes up when I'm talking to other entrepreneurs. They're interested in stories and tips and how things work. They want to know how to think, not how to copy (p. 977). The wrong question is: "What inspired that idea?" The right question is: "How did you

know when an idea was right? (p. 845)" Or even more specific: "How does one know whether a print ad is working?"

Your best bet for success is to treat all your decisions as empirical tests. Confidence and experimentation are not contradictory.[1012] Try anything, measure everything, and follow what works, even if that means changing everything (p. 193).

Then maybe you can be lucky too.

Chapter 86:
Invention is Drudgery

"You've got boring at 11:00, pointless at 1:00,
and interminable at 3:00."

Edison spent 18 months on the drudgery of trial and error to produce the first workable light, which lasted only 13 hours before the carbon fiber filament would burn out. He then spent another 18 months on the drudgery of trial and error before discovering that a carbonized bamboo filament would last 100x longer, and finally a practical, salable light-bulb was born.

> *"I have not failed 10,000 times—I've successfully found 10,000 ways that will not work."*
>
> *—Thomas Edison*

Isn't this true in all walks of life, not only engineering? Musicians spend almost all their time practicing in spaces (in)sufficiently isolated from other humans, landing gigs with a slurry of begging and assurance, transporting and setting up stages, and hardest of all, getting more than four people to show up.

In the audience, we enjoy the show, blind to the hundred hours of toil behind each visible hour of glory.

Having skipped to the last page of other peoples' book, we forget that they took the journey of the whole book. So we feel bad about ourselves when we're only on Chapter Four, having already toiled quite a lot thank you very much for asking, and when exactly are we going to get to the good part?

All we seem to do is drudgery—fixing the bugs that we were sure our unit tests proved couldn't exist, tying off the loose ends of development that never stop arising, planning (p. 1065) and estimating and communicating and post-morteming, taking the wrong path and backtracking, planning the launch and the training and the targeting (p. 1211) and the positioning and the alpha testers, and losing five

hours solving a problem that on another day with different luck might have taken five minutes.

It's hard, it sucks, and sometimes the finish line isn't visible enough to still be compelled to do it, day after day. A book has a clear end; products don't. It's often unclear whether it's smarter to continue or to quit (p. 159). It's chaos whether we're on the right track or not (p. 429).

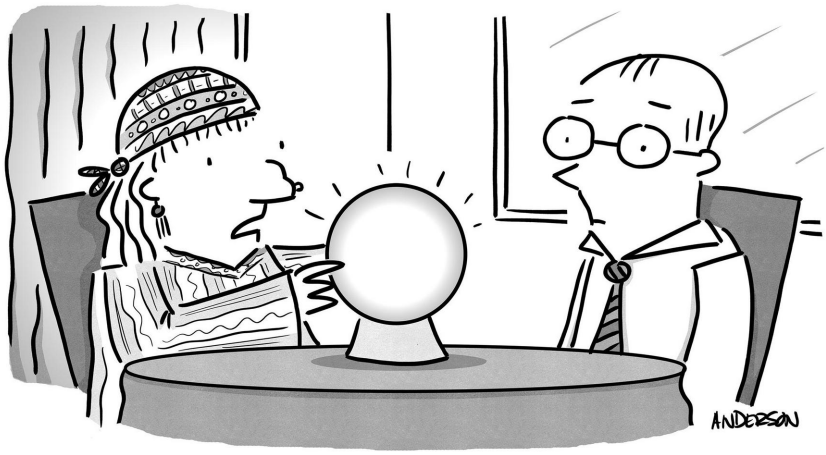
So sometimes, we quit. We quit the job thinking wrongly that the next company will be so much better. We quit the startup because it's apparently not working. We just get tired, and why shouldn't we be tired and why shouldn't we quit? Life's too short and all that.

But the ones who eventually succeed are the ones who plodded through the 5,999 filaments that didn't work, put in the 10,000 hours master a craft, and fought through the overwhelming pile of challenges and drudgery that is always required to create something great.

Afterwards, when you're panting from exhaustion and laughing because it finally worked, you'll be able to look back and say, "I really *did* something."

And that's the cue for others to dismiss it as mostly luck (p. 1035). The most annoying part is: They're not altogether wrong.

Maybe it's crazy to do all this for a fleeting feeling of accomplishment. On the other hand, maybe it's the meaning of life.



credit 1014

"Nope, nothing interesting in any of your previous lives either. You're consistent, I'll give you that."

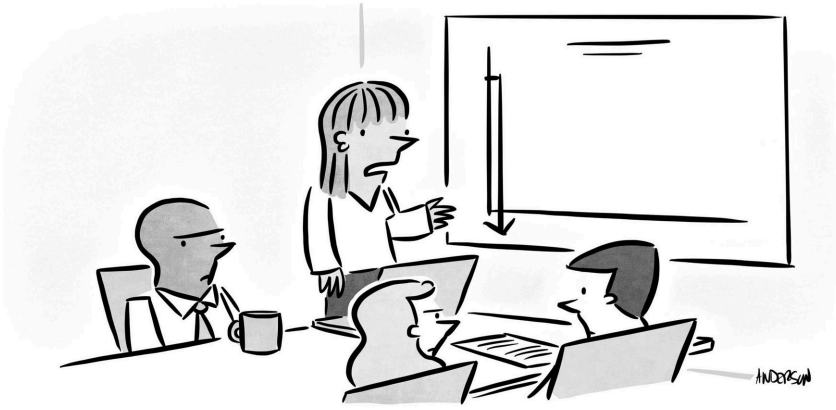
Chapter 87:
The "errors" that mean you're doing it right

> *"If you don't make mistakes, you're not working on hard enough problems."*
>
> *–Frank Wilczek, 2004 winner of Nobel for Physics*

Intellectually we know that failures (p. 1261) are inevitable when we're striving, growing, and learning. In practice we're not always so understanding when it comes to our teams, our revenue targets, and especially when it comes to flogging ourselves (p. 457).

Indeed, not all errors signify progress. Some are negligence, or just bad luck. We don't always learn from failure; in fact, sometimes there's nothing in particular to learn.

The following "errors" are the natural by-product of good decisions, or the result of a fundamentally positive circumstance that is



credit [1015]

"OK, lesson learned. Going forward no one listen to Don."

attended by the proverbial "good problem to have." Most demand a response, but they should be regarded as a necessary side-effect of success, and celebrated as such.

Re-adding features/bugs you removed from the backlog

If you're not adding back feature-requests or bugs you cleaned out of the backlog, you're not cleaning out enough.

Backlogs grow without bound unless they are culled. 1000 tickets is the same as 100 tickets, except that you haven't identified which 10% are most important. Which means you're definitely not working on the most important ones. But if you delete things, it will sometimes turn out we needed to do it after all. That's a sign that you're handling your backlog well. (In part for this reason, you should have multiple backlogs (p. 711), except for the work you're doing right now.)

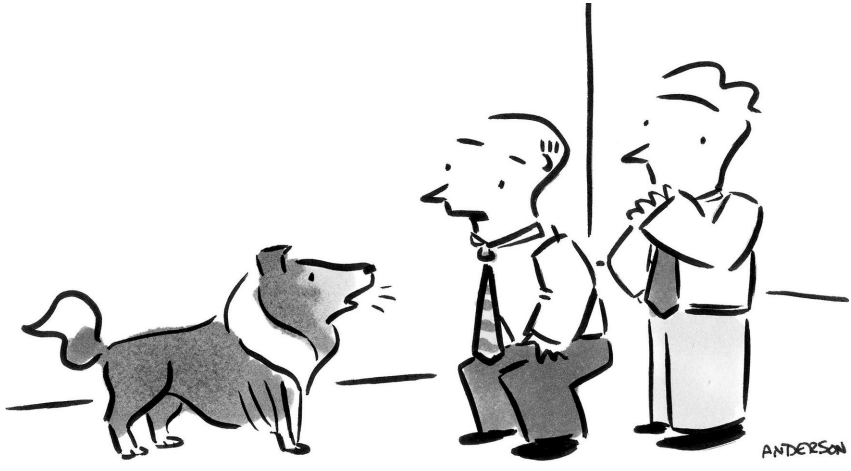
Pivoting a strategy just after creating it

A strategy that never changes is wrong, and the most likely time to discover that it's wrong is just after you wrote it down, because you have the least practical experience with how it intersects with real life.

If you're not pivoting a strategy that turned out to be wrong, you're penalizing the company with months or even years of useless work, followed by rework (if you haven't run the company into the ground), putting the entire future of the company at risk.

Great strategies (p. 489) are hard to create, and released with great fanfare: Sparkling documents, inspiring presentations, pulpit-thumping speeches, reorganized teams and strategic-pillarized work. So the last thing you want, is come back in a month and say, "just kidding, we were wrong about something important."

Will you lose credibility? Will anyone believe the new strategy? Will people think "management doesn't know what it's doing?" These are risks you have to take, because executing the wrong strategy is far worse. Indeed, this is the expected result of a new strategy; it's highly



"What's that boy?! A paradigm shift?!"

credit[1016]

unlikely you got everything right the first time. The best way to communicate, is to say everything in this paragraph out loud, so everyone knows that you know that they know, and that you're putting the company first, and ensuring that no one is doing work that we secretly know is the wrong work.

Refactoring infrastructure after growing 10x

If you're not refactoring your infrastructure after a tenfold increase in growth, you over-engineered your original infrastructure.

Having scaled WP Engine to millions of websites serving tens of billions of requests daily, I can tell you that scale is hard (p. 773), and that you don't know what will break under scale until you're already scaling (because you have to handle things that you can't even measure yet (p. 1345)). If you over-engineer your original product, you're simply not shipping your SLC fast enough (p. 101), and your naïve attempt at engineering massive scale from the start is just another form of premature optimization.[1017]

Adding words because messaging was too terse

If you're not adding back words because the messaging was so terse that it became confusing, your marketing is too verbose, too fluffy, and probably doesn't know what it's trying to say.

People don't read. Tweets are short. Google Ads are shorter. Email titles are shorter. People bounce off home pages in three seconds. No one reads the paragraph of text in the dialog box. You're not even reading this paragraph.

It's 100x more likely that your messages aren't punchy enough, aren't specific enough (p. 317), than that they're so brief as to be unintelligible. Nowadays you can use ChatGPT prompts to get you 80% of the way there, so you have no excuse.

credit 1018

Adding back features you removed

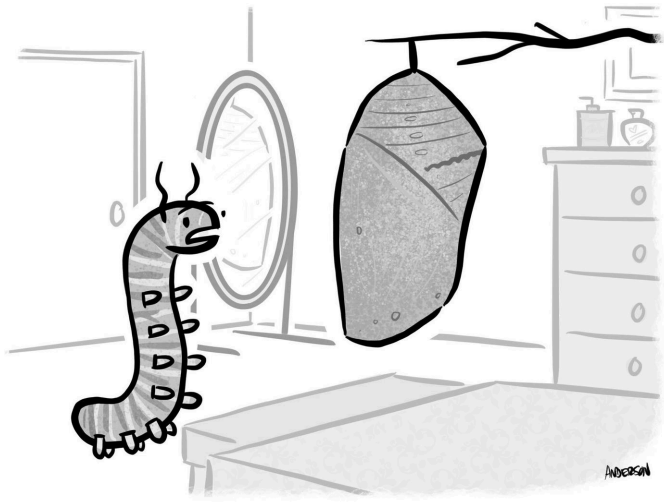
If you're not adding back features you removed, you're not removing enough.

Many products never remove features. This indicates we're not being critical enough, not weeding our garden, not learning what customers really want (p. 239), not understanding what's useful, not admitting when we got it wrong, not shifting when the market shifts. When we do remove a feature, sometimes it will turn out the feature really was important after all. While of course in a perfect world we wouldn't have made that mistake, it's a natural consequence of weeding.

Fixing lots of bugs just after a major release

If you're not fixing bugs due to releasing quickly, you released too late.

While releasing garbage is a bad policy (p. 101), it's also bad to wait until "everything's perfect." Windows 95 shipped with tens of



credit[1019]

"Could you hurry it up, sweetheart? We're going to be late!"

thousands of known bugs,* and was heralded as one of the greatest software innovations and most successful product releases of its time. Contact with customers lets you know which bugs are more important to fix next, and always reveal new bugs that you weren't going to find on your own anyway.

Waiting too long to scale support or sales

If you held onto support and sales for too long, rather than hiring a team, you learned a lot about your customers, and a lot about how to do Support and Sales.

It's a classic funded-startup mistake to scale out either of these organizations too soon. Without a system in place, with materials, knowledge bases, and scripts, new hires don't know how to do the

* There were so many, there's an entire book[1020] explaining how to work around 1000 of them.

job. A distributed-work environment makes this 10x more challenging. The second you put someone else between you and a customer, your pace of learning and understanding falls off a cliff. Wait until it's breaking for lack of scale,* then scale.

Letting someone go soon after hiring

If you held onto someone even though you knew isn't was never going to work, you're doing a great disservice to that person, and your team, and your company, and yourself.

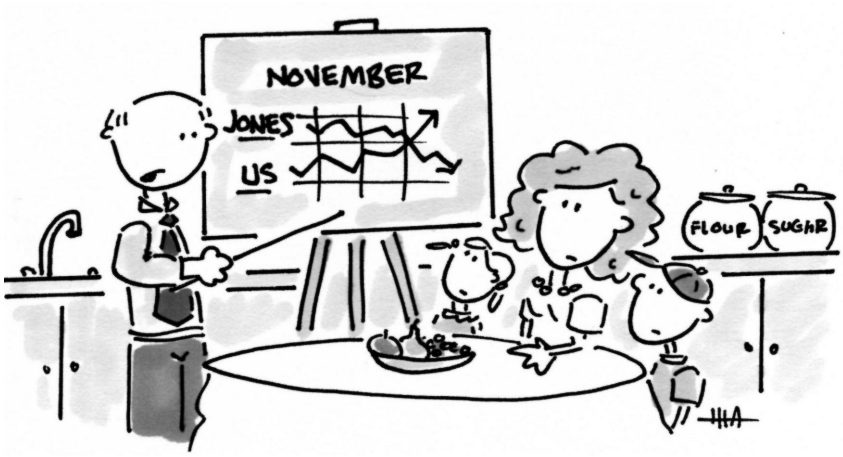
Of course this shouldn't be done capriciously, but no one benefits from dragging it out. One likely outcome is that you lose good people, because they see you building a team they don't want to be a part of. Another is a deluge of meetings, complaints, side-conversations, and general worsening of morale. And lower productivity, as competent people cover for the incompetent. You have to face the truth (p. 657) and act quickly. If this is happening a lot, it's also urgent that you fix your hiring process; in the meantime, the rule still applies.

Ignoring a competitor's move that turned out to be important

If you're not ignoring most of your competitor's moves, then you're playing their game, not yours.

It's essential to stay focused on your unique value proposition and not get sidetracked by every move your competitors make. For instance, when a major competitor of Dropbox launched a similar service at a lower price, Dropbox chose to stay the course, focusing on

* Once at scale (p. 773), this rule no longer applies; at that point, you're mismanaging the company



credit 1021

"As you can see, we've not only kept up with the Jones, but surpassed them. Next up, the Nelsons."

their superior user experience and brand loyalty, rather than engaging in a price war.

Sometimes it will turn out that you really do need to react, but that signal will come from customers, in the form of them asking for features "because so-and-so has it" or cancelling and going to a specific competitor. That indeed demands a reaction, but only because you're seeking what's genuinely best for your customers, not because you're reacting to everything that competitors do.

Rejecting a lucrative, distracting deal

If you're not rejecting lucrative deals that don't align with your strategy, then you don't have a strategy (p. 489). If you're not rejecting relationships that don't align with your core values, you don't have core values (p. 827).

Money is too tempting to reject. Money is one of the main reasons you're building a company in the first place. Money is what keeps the

doors from closing and enables the next set of things you want to do. It's even wise to say "yes" instead of "no," (p. 621) so long as there's enough money in it.

But money is not more important than strategy, and it cannot be more important than your values, otherwise you're saying that you don't actually have either one. There's always a way to make more money—a different product, different industry, or breaking the laws or being unethical. There's a reason why you're taking the path you're currently on.

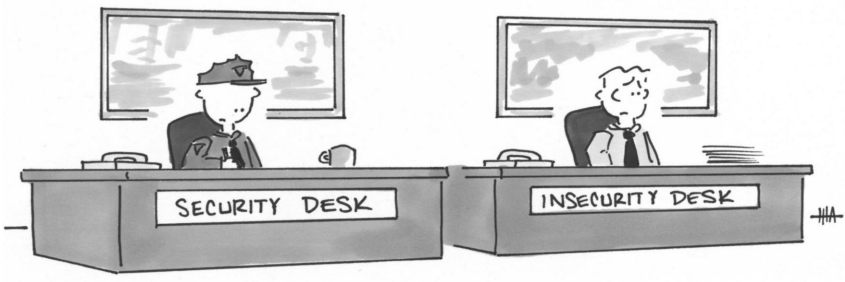
Not all problems are indicative of poor decisions. It's easy to be hard on ourselves, but sometimes we should do just the opposite:

Celebrate our devotion to good decisions and good strategy, even when they have negative consequences.

That means you *have* a strategy, and have the ability to make the tough, wise decisions.

HT Hassy Veldstra[1022] *for finding the Staedtler advertisement.*

Chapter 88:
Scars



I have a scraggly patch of hair on my right calf from when I scraped off a swath of skin in an Ultimate Frisbee tournament. (I made the catch for the score. My wife asked "Was it worth it?")

We all have scars. The interesting ones aren't physical, and are more subtly revealed. It's the guy who's a little *too* steadfast in his claim that "VCs are evil." It's the founder who says she doesn't need to talk to customers before embarking on a $200,000 development project because "customers don't know what they want until you show it to them." It's the developer who is sure that "Java sucks."

Scars are part of what makes us unique.

Dwelling on our peculiar trauma is a comforting way to develop that uniqueness.

> *"I like scars.*
> *I want to remember.*
> *I want to feel blood and tears.*
> *I want it to feel tender.*
> *I'm watermarked, just like forever."*
> *–Matt the Electrician,*[1024] *Home*

That uniqueness is good, or so says most advice (p. 543). Embrace your scars, embrace your identity, own it completely, and suddenly you've solved one of the key riddles in life, not just in startups: Who am I (p. 1433)? What should I be doing (p. 399)? How is that different from who anyone else is? How do I communicate this (p. 939)?

Some of my baggage, however, is a hindrance, whether or not it's also busy "defining me." I still find myself sometimes running WP Engine[1025] like the bootstrapped startup that it was for the first

18 months of its life, instead of the funded growth machine that it evolved into.

For example, last week I spent about 10 hours saving about $1000/mo in hosting costs. Not bad, you say, that's $1000/mo right into your pocket! You can even make a financial argument: That's 10 hours to earn $12,000 over the next year, which means my time was worth over $1000/hour, and that's a good hourly rate (p. 1413) no matter what.

No, not no matter what. I could have used those 10 hours to make it easier to share a website speed report,[1026] and that might have resulted in 1000 people trying it over the next year, 10 of which end up moving their WordPress site to our platform, and a few consultants who collectively put 20 of their clients on us. Even at $50/mo, that's 30 × $50 = $1500/mo in direct new revenue plus side benefits in marketing and branding.

If you're bootstrapping, getting that $1000/mo *right now* is the better choice (p. 353). Money is scarce, time is precious, and $1000 today is better than $5000 next year (which you might not survive to see).

But if you do have money in the bank it's just the opposite. The whole point of an investment (p. 867) is to turn "money today" into long-term value, meaning a growing, profitable, predictable revenue machine.

So even if you know your scars, embrace them, and have perfect rationalization of why every decision is the correct one for you, it *still* might be wrong.

What can you do to mitigate this?

First, you have to decide what your core beliefs are: The inalienable tenants that you want to live your life no matter what—even if it means making less money, or increasing risk, or hurting your brand, or hurting your feelings, or limiting your fame, or closing doors, or losing relationships. The things that you believe so strongly, that if a relationship were contrary to it, then it's the relationship that is wrong, and thus it's good that it dissolves; that if making money a certain way

is contrary to it, then you don't want to make money that way. These are your core values, and also form the basis of the cultural values of your company. This is who you are (p. 1005).

Second, you ensure your startup journey (p. 861) conforms to these values, as well as your natural strengths and weaknesses. Decide to build a company in which the correct, consistent decisions are the ones you'll naturally take. If you love optimizing the last dollar out of the process—as I apparently do—build a bootstrapped company. If you don't like working with people, be a Micropreneur like Rob[1027] and Patrick.[1028] If you want to leave a mark on the world, have big ideas with lots of people and lots of money, seek advice from those who have walked that particular path before you, who help you become a better version of yourself (p. 751), rather than pushing you to be something "rational" but unnatural.

Third, use this as a guide for interacting with the rest of the world. Having defined "yourself," and made decisions consistent with that, you have to find out where you are wrong, where you're ignorant, where you're doing the wrong thing according to your own definition of "right." You can't do that alone, almost by definition; otherwise you would already have fixed the problem. So do what I'm doing now: Surround yourself with trusted advisors and be completely and continuously honest with them, then actually listen and learn. I know I'm naturally a bootstrapper, a "get to the first $10M in ARR, but only after seven years, and then what?" type of person, so I know I need constant course corrections.

Escape the echo chamber of your own head. Because, as in the example above, you **cannot know you're wrong**. Not even if, like me, you're proactively introspective (p. 657), escaped the clutches of Impostor Syndrome (p. 457), and have a few successful startups under your belt.

So embrace your scars, let them lead you to who you are and what you believe, construct your life, relationships, and organizations around those, but then with that solid foundation, seek (compatible!)

guidance on how you can execute on that despite the weaknesses and blind spots and flaws that those same scars create for you.

You are your scars.

But you're also more than that.

Chapter 89:
Why it's nice to compete against a large, profitable company

"Oh, that's the looming specter of dwindling profits.
Just ignore him."

credit 1029

A big, profitable company seems like the hardest thing for a small company to compete against. They have everything: money, brand, momentum, existing customers, press, product teams, distribution channels, expertise, market insight, analysts, sales offices, product features, and, scariest of all, a working business model.

All a little startup has is a decent idea and extremely greasy elbows.

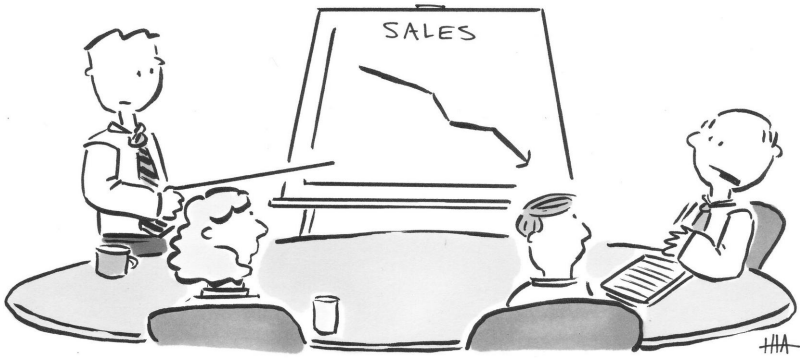
But David has a clear path to slaying Goliath. The insight is: **The profitable revenue stream is a prison.** It's the Achilles heel that allows the little guy to win.

A company with a large, profitable, growing revenue stream betrays facts useful to a startup:

1. There's a huge market (p. 71) to be had (else it wouldn't be large and growing).
2. This market is willing to pay (p. 275) far more than cost for this product (else profits wouldn't be generated).
3. This abundance will last for a while (large, profitable businesses typically die a slow, sagging death rather than disappearing in a flash).

This means the market is ripe for an Innovator's Dilemma (p. 419)-style disruption. A startup with new cost structures, new technology, and new ideas can be competitive at $\frac{1}{2}$ or possibly even $\frac{1}{10}$ the price, even if the product is merely just-good-enough.

But wait! The big profitable company can just lower prices, thereby removing the main competitive advantage from the upstart, right? Wrong. The big profitable revenue stream is the goose that's laying the golden eggs. The goal of a large company is to *protect the profit stream at all costs*, even if that means giving up on innovation or speed. The current valuation of the company is based on continued growth of revenue and earnings, not erosion due to ankle-biters. Watch how fast your stock plummets when Wall Street thinks your future earnings are in jeopardy.



"Couldn't we just move the bottom line down a little?"

Don't forget: Small changes in top-line revenue create massive changes in profitability. A business with a 20% profit margin is very healthy. If you lower top-line prices by 20%, your costs don't magically decrease 20%, so now your profits are 0%. So if a startup cuts prices by 50% or 80%, the big company cannot chase. In fact, even reducing top-line by a measly 10% still cuts profits in *half*—a penalty too massive to endure, for an effect (slightly lower prices) which won't materially change the price conversation amongst customers.

Therefore, a large company asks: "If we're going to lose on price to the low end of the market, how can we charge *even more* on the top end, to make up for that lost business?"

Will that strategy work? It might! Either way, the new startup can grab $\frac{1}{2}$ of that big company's low-end market share and still be profitable, because it started with a much lower cost structure, with new ideas, new tech, and a new business model.

But wait! Perhaps the big company will sacrifice earnings for growth? Not anymore. That's a young company's game. In the big-boy and big-girl world of real, at-scale companies, valuation is about total future earnings. Growth is important only because it leads to more earnings, not because it's "growth for growth's sake." That's the

argument a young company uses, when the primary goal is to become dominant in a market before someone else does, setting up decades of future profitability.

A final word of caution. All this applies *only* if you're attacking the product line that generates the massive profits. **If you're attacking a loss-leader, the situation is reversed.**

Big, profitable companies often have other lines of business which are unprofitable, sometimes extremely so. The profitable business unit funds the others. For example, Google's profitable search business funds GMail. Amazon's retail business funded AWS, and now that AWS is closing in on $10B in annualized revenue with 20% profit margins, it's funding other projects as we speak.

Attacking a profitable business on its loss-leaders is a terrible strategy, because it can use all its powers against you, plus orders of magnitude more dollars, and not care about a direct business model to support those decisions. *That* is a scary competitor—lots of resources and nothing to lose!

For example, Microsoft decided to make Internet Explorer a loss-leader against Netscape, and destroyed that company. That began only after Netscape was large and winning, so this wasn't a strategic error on Netscape's part, but rather a clear demonstration of the power of a profitable company who doesn't care about making money in a certain market. On the flip side, Google built a $1B business applications product line ("Docs") that competes against Microsoft, because in this case "Office" is the profitable line of business that Microsoft can't impinge.

So, competing against a large, profitable, growing business might be the smartest thing you can do! Just make sure you're hitting them where they're fat, not where they're able to beat you with their size.

Chapter 90:
Quarterly strategic planning using the fairytale structure

FAIRY TALES · NARRATIVE · STRATEGIC OBJECTIVES
OBSTACLES · MAIN ACTIVITIES
METRICS & INDICATORS · NOT DOING
FULL EXAMPLE: TESLA

"You ate some porridge, broke a chair, the bed thing's a little weird, but I'm thinking 3 months community service, maybe a fine. You should see my other case. Two kids chow down on a witch's house, then burn her alive in her own oven."

credit 1031

Using the narrative structure of the fairytale, we can execute the classic "Double-Diamond" design pattern, driving our analysis, decision-making, actions, and our final exposition, in an intelligent, systematic way that everyone can easily understand.

It's going to look like this:

1. **Objectives:** What are we trying to get / what happens if we don't?
2. **Obstacles:** What stands in our way?
3. **Actions:** How do we overcome the obstacles while advancing our objectives?
4. **Measures:** How will we know we've been successful?
5. **Not Doing:** What important things have we chosen not to do, so that we can achieve the above?

Now let's tap into humanity's collective unconscious, and see exactly how to run this framework in practice.

THE FAIRYTALE CHEAT CODE

The fairytale structure is thousands of years old, crossing all peoples of the globe, a reflection of a deep, shared humanity.

You've heard it your entire life, from centuries-old childhood stories to every Pixar film[1032] to advertising[1033] to the news, so you need only a reminder, not an explanation:

1. Once upon a time there was *an unsuspecting hero*.
2. Every day, *the way things are*.
3. But then, one day, *change/event that motivates action*.
4. And so *adventure begins*.
5. It was almost impossible because *challenges/obstacles*.

6. Until finally *victory*.
7. And forever after, *the world is different*.

Fairytale structure is a cheat code for effective communication, as journalists have exploited for years. For example, this simple "case study" that I've used in sales calls at WP Engine takes less than a minute to tell:

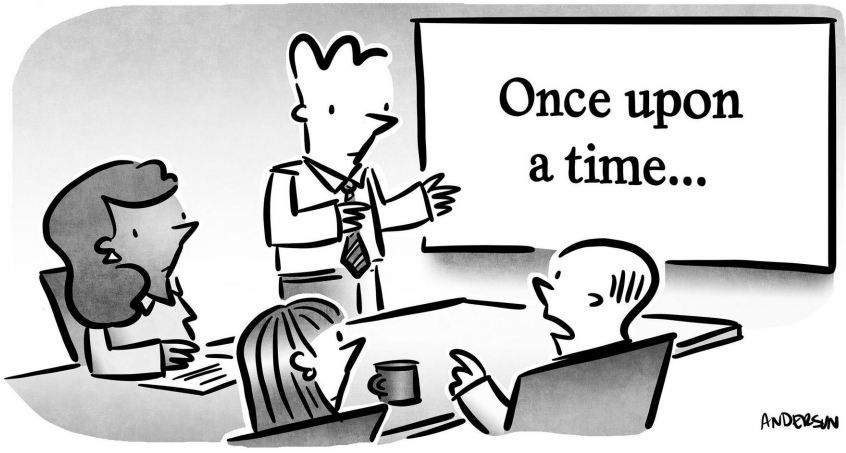
> One of our customers is _____. Like you, they are a popular food blogger, making money through advertising and occasional affiliate product sales. One Thanksgiving, on their busiest day of the year, their site crashed because of all the traffic; they missed out on thousands of dollars of revenue and hundreds of newsletter sign-ups. They moved their site to us, and that never happened again; they just focus on their work, build their brand and business, and never worry about technology.

It's equally effective for presentations, fundraising, and product proposals—a ready-made outline, resulting in a straightforward narrative structure. **Our brains are pre-wired for fairy-tales**, so we naturally follow the presentation as it flows into those primordial ruts in our gray matter.

A less obvious but extremely powerful application of this principle is in creating a **strategic plan**, i.e. **analyzing the current situation and deciding what needs to be done**. For a single team, this means being in command (p. 413) of the most critical things that need to be done right now. For a department at a larger company like WP Engine, this is how I build our quarterly strategic plans.*

Over the past two years, together with a few product teams at WP Engine, we've created and honed a simple but effective framework for the **Fairytale Plan**—strategic work-planning that has worked

* It looks like I'm just slapping the word "strategic" on everything, to make it sound important. Indeed, people often do that. As you'll see, the purpose of this is to connect work to the strategy, so it really *is* strategic planning!



credit[1035]

"Listen, I understand about the burden of exposition, but can we just see the 4th quarter numbers?"

well both to drive the process of figuring out what the plan should even be, as well as communicating the result throughout the company. The process is akin to the widely-admired Double-Diamond design process,[1034] and the final result matches the fairytale structure.

No wonder it works, and resonates with everyone.

Here's how you can do this too.

THE NARRATIVE STRUCTURE OF THE STRATEGIC FAIRYTALE PLAN

Here's how our strategic plan will look, and the guiding questions that help us create each section.

	Strategic Plan Narrative	Guiding Questions
Objectives	**We must advance our strategy by achieving these objectives.** *Fairytale* 1. Once upon a time... 2. Every day...	• What is happening now? • What do we want? • What happens if we don't get it? • If we weren't encumbered by roadblocks, with no distractions and no interruptions, how would we advance our strategy right now?
Obstacles	**But obstacles stand in our way.** *Fairytale* 3. But then, one day... 5. Obstacles*	• What is standing in the way** of "what we want?" • What makes it difficult to achieve the objectives?
Actions	**So here's what we're doing.** *Fairytale* 4. Adventure begins... 5. Tackling obstacles...	• How do we overcome the obstacles... • ...while advancing the objectives? • ...with the people and capabilities and strengths we currently have? • What changes need to be made, so that the answer becomes obvious?
Measures	Here's how we'll know **whether we're making progress**, and **when we're finished.** *Fairytale* 6. Until victory... 7. The world anew.	• How will our company be different? • How will we know, objectively?

* In a fairytale it's fun to surprise the hero with obstacles. Our plan must do the opposite: Clarify the obstacles up front.

** There are many kinds of obstacles: internal execution challenges, interruptions, dependencies, competitive pressure, market dynamics, just to name a few.

	Strategic Plan Narrative	Guiding Questions
Not Doing	**Important topics and ideas that we're not doing** (yet).	• What do we really *want* to do, but have to wait so that even more important or urgent things can be done? • What did we leave out that is controversial?

There are techniques and pitfalls for each of these sections. Here's how to navigate it:

STRATEGIC OBJECTIVES

Narrative Plot Point
We must advance our strategy by achieving these objectives.

Guiding question
If we weren't encumbered by roadblocks, with no distractions and no interruptions, how would we advance our strategy right now?

Why can't we skip this section?
If you run very fast in the wrong direction, you fail. This names the right direction.

Objectives shouldn't summarize the entire strategy; your strategy document (p. 489) already does that. Instead, link to that document for reference, and highlight 1-3 specific objectives as the subset of the strategy that make sense to advance right now.

credit 1036

"I think I see why we're not getting anywhere."

"Right now" is the difference between this *plan* and your *strategy*. The strategy is the long-term vision and the main ways that you will achieve it. The plan is where you select the specific next steps that materially advance the strategy while achieving results that you need today. "Results today" is typically revenue growth, cost reduction, or some internal transformation.

Later in the process of creating the plan, when you're determining *how* to achieve these objectives while also dealing with the obstacles, you might get stuck, unable to invent anything that works. In that case, you might need to choose different objectives. Or you might decide that "eliminating obstacles" is all you can accomplish right now—unfortunate, but useful to declare if true. During the planning process you should expect and even encourage fluidity between exploring the problem-space and solution-space.

The specific objectives will evolve over time, but it's OK if they're the same for a few quarters in a row. Indeed, that gives the team stability and focus. How do you know if you're sticking with an objective for too long? Later you'll be asking the question "how will the world be different." If there are good answers, then we'll know whether we're making progress, despite the objectives remaining the same.

OBSTACLES

Narrative Plot Point
But obstacles stand in our way.

Guiding question
What is preventing us from making rapid progress on the objectives?

Why can't we skip this section?
Often the most important business challenges appear here, not in the objectives. This is where you face difficult truths about your current situation, which will ensure the work you select will actually overcome them.

> *"If I had an hour to solve a problem, I'd spend 55 minutes thinking about the problem and 5 minutes thinking about solutions."*
>
> *–Albert Einstein.*

I agree with Einstein (but then, who among us is in a position to argue against Einstein?). This is the most important section, and often where you'll spend the most time.

You'll be tempted to ignore Einstein, but it is crucial that you do not. You'll want to jump right to the solution. You'll say you've al-

ready talked about this stuff, and you know what to do. You'll say you don't want to rehash all our challenges; let's cut to the action.

I understand, I feel the same way. I want to activate on any reasonably-good idea right away. But without fully exploring the problem-space, you're not in a position to even know the goal of the solution-space. You'll just keep repeating mistakes, keep avoiding the difficult truths (p. 657), and you won't improve.

It's important to generate many options first; only by exploring the whole space will you uncover the things which are the *most* important to solve now. Your team will thank you for the clarity, not only about what they need to tackle, but what they're allowed to leave alone, even though it's on fire.

Here are some types of obstacles you should explore, in no particular order:

Difficulty executing the strategy: "Why is this hard?"
Executing the strategy might be intrinsically difficult, like building something we've never built before (thus maximally risky and uncertain), or having to create novel algorithms, or having to deploy infrastructure at scale (p. 1345), or pivots, rewriting, rebranding, or repositioning that will take intense effort and coordination. Whatever makes our strategic goals difficult or risky, is an obstacle that is definitionally strategic to remove.

Team challenges: Humans are hard.
Do we have the right skillsets on the team? The right experience levels? The right motivation? Are they burned-out? Is there enough people to do all the work?

Competitive pressure
Is a competitor winning business in a way that demands a response, even if it's not directly related to our long-term strategy?

Customer retention
Are customers leaving at high rates,* and therefore we have to address that immediately even if it's unrelated to our long-term strategy?

Interruptions
Are we dealing with interrupt-driven work instead of planned work, such that we're unable to make enough progress on strategic work? Do we need to do work to reduce the frequency or magnitude of the interruptions, perhaps through automation or rejecting certain types of work?

Too many bugs
Are we failing on our basic promises to our customers of quality and performance, due to mounting bugs or other issues? Do we need to attack those now, even if they are unrelated to our long-term vision?

Dependencies / coordination
Is it impossible for us to execute or deploy or iterate, due to some other team or process or platform or architecture? Do we need special coordination of work in order to achieve our own goals?

Process bottlenecks
Are our tools getting in our own way? Is our process preventing throughput? Not just nice-to-have improvements, but step-changes in our productivity, and likely also our happiness?

Market evolution
Are there trends that are making our current product or positioning less effective or less relevant? Does this even mean our strategy needs to change, to keep up with the changing external landscape?

Narrow these down to 1-3 most critical issues. The guiding question is: Which are the few issues where, if we solved them or took a huge bite out of them, but allowed all others to fester without any progress whatsoever, we would have substantially improved our velocity, or competitiveness, or team dynamic, or retention rate, or something similarly critical. Whereas if we allowed those few issues to go untreated, then even if we completely solved five others, we would still be unable to achieve our strategic objectives?

---

* My rule of thumb for "what is a 'good' cancellation rate" is: 3%/mo for consumer, 2%/mo for small business, 1%/mo for enterprise. More on that here (p. 335).

If that question is still hard to answer, then as a guideline I like the following order of precedence. That is, if you have a *serious* problem in some of these categories, I generally pick whichever comes first:

1. Retention (if customers are leaving, the business model and product isn't working)
2. People (must have the right team)
3. Throughput (can't do enough strategic work per month)
4. Growth (Details on how to tackle growth challenges (p. 1191))
5. Competition
6. Health of the code base
7. Execution risk

> *"A perfect formulation of a problem is already half its solution."*
>
> *–David Hilbert*

MAIN ACTIVITIES

Narrative Plot Point
Here's what we're doing.

Guiding question
How do we overcome or side-step the obstacles, while advancing the strategic objectives, with our current people and capacity?

Why can't we skip this section?

If you don't select a few, large, impactful things to do ("Rocks"), you will wile away your time chipping away at incremental things, believing that you're making progress because you are "agile," but in fact not tearing down the obstacles nor achieving the objectives quickly enough.

While putting your finger on the crux of the challenges is often the most emotionally-difficult part, deciding what to do is the most intellectually-difficult. It's hard to think of great ideas, and no framework or process magically causes you to be creative and insightful.

As with obstacles, you should start in "generation mode," coming up with as many ideas as possible, before switching into "selection" mode, where you pick which ones to do. You should already have

credit[1037]

"Listen, you've got so many you don't know what to do, and I love the taste of children. Maybe we can help each other out."

queues of ideas from around the company (p. 711); if not, now is a good time to create some. If you're having trouble coming up with creative ideas, use Extreme Brainstorming (p. 53) to shake things up.

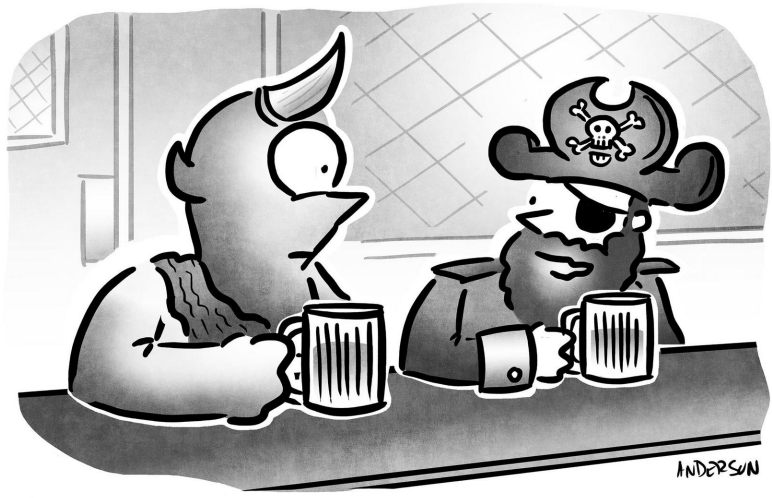
If you're having trouble figuring out how to deal with the obstacles, consider the parable of Herbie (from *The Goal*[1038] by Eliyahu M. Goldratt). A group of hikers are climbing a mountain. One hiker—Herbie—is constantly falling behind, preventing the group from achieving their goal of reaching the summit. There are three ways to deal with Herbie: (1) Leave Herbie behind; (2) Help Herbie get better; (3) Decide that you're not going to achieve your goal. These options help you think of solutions; respectively: (1) Let that problem burn or ignore certain work that you've been assuming "must" be done; (2) Do extra work that mitigates the problem; (3) Change your objectives to something you can actually achieve.

Often the best solutions come from synthesizing several ideas (p. 589). For example, often at WP Engine an idea that increases website performance for our customers also decreases our costs, because it takes less CPU time or fewer bytes transferred over the internet to accomplish the same end result. This is another reason to use things like Extreme Brainstorming (p. 53); even if an individual idea is bonkers, it might be synthesized with a practical idea into a great solution.

For the final selection process I recommend the Binstack method (p. 603), because you're picking one or just a few critical big things to do. Your objectives and obstacles act as natural filters, immediately eliminating anything that doesn't directly address those, but allowing yourself a few secondary prioritization dimensions.

You might be tempted to use an ROI analysis (p. 171), but this is the wrong framework. ROI is good for smaller activities, where the goal is to find the "best use of time." For your most critical pieces of work, the goal is not to optimize your time, but rather to maximize your impact on the most critical things.

See this Rocks, Pebbles, and Sand work-planning framework (p. 221) for a complete system that covers all kinds of work, includ-



credit[1039]

"OK, but if we work *together*... Whammo! Depth perception!"

ing these most-critical activities and medium-sized work where ROI analysis is appropriate.

> *"A successful solution to a problem makes the problem appear to have been nonexistent in the first place."*
>
> —*James Surowiecki*[1040]

METRICS & INDICATORS

Narrative Plot Point
Here's how we'll know whether we're making progress, and when we're finished.

Guiding question
How will we know whether it's working? What is changing? What did we learn?

Why can't we skip this section?
It's easier to avoid accountability, believing that hard work and high throughput is the same as having an impact. Saying "focus on inputs, not outputs," which is true when executing work but false if it means we're not actually getting the outputs we need. We must not let ourselves off the hook, but neither should be measure ourselves against metrics we don't believe in, or insist that all important things can be boiled down to a number.

The "Activities" are supposed to make progress on the objectives, but how will we know if that's actually happening? They're supposed to deflating the challenges, but are they really getting eliminated?

In the ideal case, we should measure progress with numbers that are objective, well-defined, measured daily, and have a clear "stopping-point" where we can declare victory. But we all know that most things we can measure aren't so ideal, and sometimes the most important things aren't a number.

credit 1041

"OK, we've got three sides now. I think we're making good progress."

To the extent that important things *are* numbers, use this KPI framework (p. 645) to decide what things should be measured, and as a checklist of what a "good" metric is.

For concepts that are not a number, capture their essence in writing. Make the language crisp and obvious not only inside the team but to outside stakeholders. Nowadays you can use AI to take a rambling audio description and turn it into a pithy statement. An example might be: "Be revered as the most innovative product in the category" or "Have the best social media content."

These statements can be surprisingly powerful despite not being connected to a number. Even when people disagree later whether we're making progress against them, those are exactly the right sorts of debates to have, because you're sorting out how things are going, and whether you should make a change. That's the point of metrics.

NOT DOING

Narrative Plot Point
Here's what we're *not* doing (yet).

Guiding question
What do we really *want* to do, but we just have to wait so that even more important or urgent things can be done?

Why can't we skip this section?
Decisions are about saying "no" (p. 621) even more than saying "yes." Without articulating the "no's," the team is less focused, the decisions are less clear, and stakeholders wonder whether you're even aware of the problems that you chose not to tackle.

> *"The difference between successful people and really successful people is that really successful people say no to almost everything."*
>
> *–Warren Buffett*

Deciding to do a few things always means deciding not to do dozens of other things.

"Well, not doing them *for now*" we like to say, to put the listener at ease. But actually, most of them are "not ever," because we always have 10x more ideas (p. 725) than we have time to execute. That's the

lesson of every Jira backlog, feature-collection system, or prioritization spreadsheet.

When you don't articulate what we're *not* doing, it can appear that you don't even know those things exist. Perception is reality. Stakeholders or even people on the team believe you're ignoring important things, when in fact you've *selected* certain important things. You need to say that out loud.

It's not just about proving yourself to an executive, it's about reminding yourself and your own team, especially when a big customer complains about one of the things you're intentionally ignoring, and the pressure is on, the emotions are high, and you need the resolve to stand firm on "no."

> *"You can do anything, but not everything."*
> *—Derek Sivers*

Next quarter, when you're going through this process again, this is a great list to revisit. You'll be surprised how many items seem less important now than before. And some might be more important; perhaps now is the time.

At WP Engine this a simple bullet-point list, just one line per bullet. It says what the topic is, and doesn't even explain why it wasn't selected. You could go further, explaining why each didn't make the cut. You could go still further, creating a "Now / Next / Later" roadmap.*

---

* Or, my personal preference in that three-column genre: "Now / Likely / Maybe." This better indicates not only timeline but our confidence that those are even the right ideas. After all, how many of those "Later" items do you ever get to? If the answer isn't at least 50%, then they're *not* "Later," they are merely a possibility.

Remember, this isn't a random bucket of 100 feature-requests. It's a list of things that *are* important, possibly even seem *urgent*, that you really do want to do, that maybe only barely didn't make the cut. In short, these were the difficult decisions.

> *"I'm actually as proud of the things we haven't done as the things I have done. Innovation is saying no to 1000 things."*
>
> —*Steve Jobs*

FULL EXAMPLE: TESLA

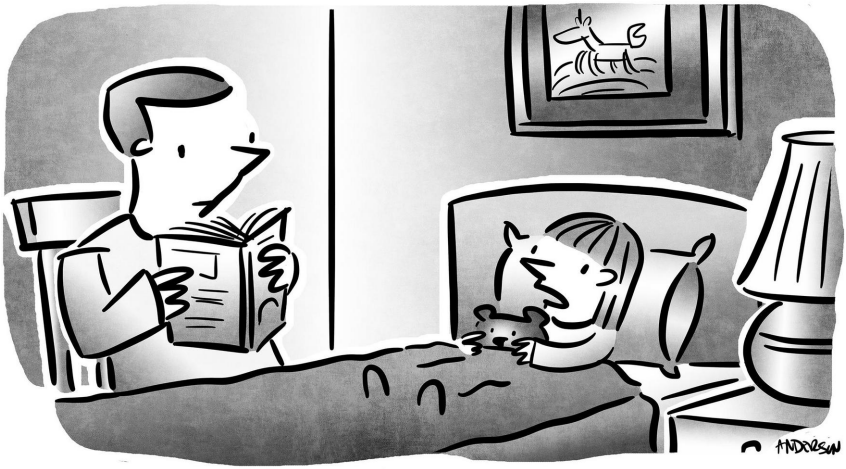
The final result is clear and simple, hiding the complexity of the process that created it.

Let's use Tesla as the example, because although it's over-used, that means you're probably familiar with it, and thus you can see how it all comes together:

Strategic Objectives

Tesla's long-term vision is to accelerate the world's transition to sustainable energy through the widespread adoption of electric vehicles (EVs).

To do that, we need to:



credit[1042]

"I'm just saying, 'once upon a time' is lazy writing, but go ahead I guess."

1. Invent vehicles that are not only battery-powered but also desirable, because of their performance and safety.
2. Create production lines that are scalable but also cost-effective, so we can be profitable.
3. Generate excitement and loyalty by bucking consumer expectations that EVs are impractical and have low performance.

Obstacles

These obstacles stand in our way of achieving our vision:

- **Intrinsic difficulty:** Inventing new types of battery and production lines is near-impossible.
- **Market Skepticism:** Overcoming consumer and industry skepticism about the viability of EVs.
- **Manufacturing challenges:** Scaling up production to meet demand.

- **Infrastructure:** Lack of widespread EV charging infrastructure creates range-anxiety.

Main Activities
We will overcome these obstacles while achieving our strategic objectives by:

- **Innovate in Battery Technology:** The insight is that the battery is the key. Develop more efficient, longer-lasting, fast-discharge batteries.
- **Direct-to-Consumer Sales:** Bypass traditional dealership models to control pricing and customer experience.
- **Scale Production:** Invest in Gigafactories to massively increase production capacity, hiring specialists and investing with the insight that factories are not just a necessary step but strategically vital.
- **Infrastructure Development:** Invest in a network of Supercharger stations for convenient charging.

Metrics & Indicators
We know whether these activities are working through:

- **Battery Cost and Efficiency:** Measure improvements in battery cost per kWh and immediate-discharge rate.
- **Vehicle Sales and Market Share:** Track the number of units sold and market share in the EV sector.
- **Production Capacity:** Measure fully-completed cars/day per Gigafactory.
- **Infrastructure Expansion:** Number of new Supercharger stations operational.
- **Consumer Delight:** People absolutely love their Teslas, calling it a magical experience and never wanting a gas-powered car again. *(not a number)*

Not doing

We are intentionally not doing these valuable, important things, so that we can focus on the above:

- **Safety and Practicality**: Although we believe eventually we will make the safest sedans in the world, first we have to solve the challenges of batteries and performance, because those are the obstacles. We know that safety is a matter of good design, but we already know from existing EVs that if the car is safe, but not performant and low-range, then it will not be popular. So we have to solve those first.
- **Affordable models**: The company will truly scale and make the impact we want in the world, when it is accessible to most people. But at first, before factories are at scale, before the bugs are worked out, before all the costs are minimized, our cars have to be expensive. So we have to start out as a luxury brand.
- **Expand to China:** The second-largest market in the world, eventually will be the largest, and the first manufacturer there might have a decisive advantage. But it is its own special challenge, and we need a popular car that works before we jump into that pool.

Tesla is often lauded for having a clear strategy, and you've heard them report the metrics above as indicators of progress or success.

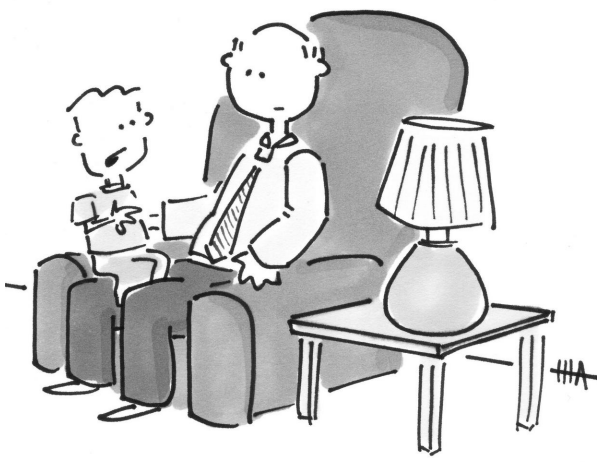
You need a plan that is just as clear. Each team needs one, and your company needs one.

Otherwise, you're probably not working on the right things, and certainly not communicating what must be done, in a way that everyone can understand and implement.

Tell your story.

Chapter 91:
SSEBITDA—A steady-state profit metric for SaaS companies

EBITDASM · TWO FLAWS · SSEBITDA
PROFITABLE GROWTH RATE · SUPPLEMENTAL



"I understand *that* part, but then where do *storks* come from?"

credit 1043

Your company is unprofitable because you're "spending to grow"—pumping money into sales and marketing faster than you're collecting revenue, resulting in accelerated but unprofitable growth. This is what you're *supposed* to do—so they say—to build a large enterprise.

Is this smart, or are you just creating a permanently-unprofitable company? How do you objectively measure whether you're strategically trading dollars-today for profit-tomorrow, or whether your business model is broken?

EBITDASM: RACKSPACE'S ATTEMPT

When Rackspace[1044] was rapidly scaling, they answered this question using a metric they called EBITDASM. Here was their logic:

> Rackspace is efficient at acquiring customers. Specifically, our CAC* is small compared to our LTV.**
>
> Therefore, we spend as much as possible on growth. We take market share cost-effectively; we burn cash in the short-run, but we're profitable in the long-run.
>
> However, we know other companies *claim* this was true for them, but they never actually get profitable! We don't want to become one of those. What can we *objectively* measure that would tell us whether we're being smart?

---

* CAC is the Cost to Acquire a Customer, defined as all Sales and Marketing expenses, divided by the number of customers yielded by those activities.

** LTV is the LifeTime Value of a customer, meaning the total gross revenue the customer will generate over the years it remains a customer. Side-note: I don't believe in LTV (p. 1355).

> Well, growth primarily comes from S&M* spend. So, what if we stopped S&M spend? We'd stop growing, but what would happen to the rest of our finances?
>
> Specifically: Would we be profitable? If the answer is "no," then our core business model isn't profitable. But if the answer is "yes," then we really are justified in saying that our underlying business is profitable, and our additional spent beyond that is indeed "to grow."

They called the resulting metric **EBITDASM**—an extension of the typical American accounting metric of EBITDA,** where in addition to the usual exclusions, we also exclude Sales and Marketing.

Computing using the usual metrics:

```
EBITDASM = EBITDA + SM
```

For small software companies who don't have purchases to amortize, don't have assets to depreciate, and don't have interest on loans, an even simpler version is just:

```
EBITDASM = [your definition of "profit"] + SM
```

I like this model, and ten years ago we were using it at WP Engine.

---

* S&M is Sales and Marketing. Why, what did *you* think that stood for?

** EBITDA[1045] is Earnings Before Interest, Taxes, Depreciation, and Amortization. It measures a company's normal business operations by stripping away non-operational expenses. It's useful for comparing operational profitability across companies with different capital structures, tax situations, and accounting practices.

TWO FLAWS IN EBITDASM

While EBITDASM is a good start, it suffers from two flaws:

(1) Halting Sales & Marketing does not mean revenue remains flat.
An assumption above was: If we stop S&M spend, the company will be static, with revenue neither growing nor shrinking. But that's not true, because revenue will fall (due to cancellations and downgrades) or grow (due to upgrades).

The intent was correct—measuring profitability assuming the company is neither growing nor shrinking*—so we need a metric that actually measures that.

(2) What if marketing and sales isn't efficient?
Another assumption was: Because we're efficient at acquiring customers, it's logical to spend as much as possible. But what about companies where that assumption is false or unclear?

How efficient does customer-acquisition have to be** before we're allowed to use this metric? We shouldn't have to debate that; the metric should apply to all companies.

Of course Sales and Marketing cost-effectiveness is important! But it's a separate metric which, by the way, you can already compute. Let's not cram multiple ideas into a single metric.

Is there a way of measuring whether the company is "fundamentally profitable" while avoiding these two flaws?

---

* To see why: Consider a company that's "profitable" in the sense that revenue is greater than costs, but it's shrinking every month. That's a company that will soon be dead, and sooner will be unprofitable. This is not what we mean by "profitable."

** There is prior art on this question, e.g. some say the threshold is an LTV/CAC of 3, because of one blog post by David Skok[1046] more than ten years ago. But others say 5 while others say 1, and I say that LTV isn't the right way to think about it anyway (p. 1355), and that you ought to use "payback period" instead.

Long-time readers might at this moment have the flash of realization that I had when first considering these flaws—it involves the same set of SaaS metrics that come together neatly in the definition of COC (Cost of Cancellation) (p. 1361).* It turns out that **COC is the key** to this metric of "underlying profitability."

SSEBITDA: STEADY-STATE PROFITABILITY

Reframing the question leads us to a simple conclusion.

Let's define a metric closer to our original intent: "Steady-state profitability," which I abbreviate as **SSEBITDA**. Longer: *How profitable would we be, if we were spending only enough to maintain the current state of the company, neither growing nor shrinking?*

Having read the previous article on COC, the formula is simple:

```
SSEBITDA = EBITDA + SM - COC
```

In short, including the S&M costs needed to replace canceled customers, but no additional S&M costs.

This solves both flaws because:

1. We've added back the cost of maintaining flat revenue ("steady-state").
2. COC includes sales-efficiency, so it works for all companies.

---

* Briefly: COC is defined as the cost of replacing customers that cancel, and thus the cost to remain at this "steady-state." Most simply it is: `CAC` × `C`, where `C` is the number of customers that cancel in a month. See the article for the formula, derivation, and discussion.

Thus it is the formula for "profit if we maintain the company as it exists today."

COROLLARY: PROFITABLE GROWTH RATE

Another interesting metric falls directly out of SSEBITDA—"Profitable Growth Rate," i.e. the rate of revenue growth the company can self-fund while still being profitable.

Where `p` is the cost of a dollar of revenue (also defined in the COC article (p. 1361)):

```
Profitable Growth Rate = SSEBITDA / p
```

Justification: If we recycle 100% of the funds from steady-state profitability back into growth, we spend `p` dollars to earn each dollar of new recurring revenue.

Any growth larger than that will require being unprofitable, by a known amount. This is handy, because **now we can justify "spend to grow" with precision**.

For example, if SSEBITDA is 12% and `p` is 4, the company can grow at 3%/mo using its own money. Supposing the company is willing and able to spend more to grow at 8%/mo, it will be unprofitable by 20%.* A company in exactly this situation is indeed "spending to grow" in a responsible manner, and will result a profitable business once the dust settles.

---

* Because 8% is 5% more than we could grow based on our own profits, and that 5% is bought at a cost of `p`=4 dollars to earn every 1 dollar of revenue, so 5% × 4 = 20%.

ADDITIONAL THOUGHTS ON SSEBITDA

Having used SSEBITDA for some years now, here are some assorted additional thoughts:

Negative SSEBITDA

A negative SSEBITDA isn't necessarily a problem. It depends on the context and goals of the company.

For example, an early-stage high-tech SaaS company will be spending much more in R&D *as a percentage of revenue* than a later-stage company. Or a mid-stage company (like WP Engine) might have higher G&A spend (office space, finance, legal, HR) *as a percentage of revenue,* until it grows into those services. Those are expenses that eat into EBITDA and thus SSEBITDA, but that doesn't mean the company won't be profitable with larger scale.

Removing other costs for a more precise "steady-state"

Or: Why *not* to do that.

It's tempting to point out that not all engineers are working on maintenance; perhaps in steady-state we wouldn't need to add so many new features, and thus engineering expenses would be lower too. And we might do away with some of that overhead expense.

This seems logical, but I don't think it's actually correct, nor worth your time to calculate. First, it's a lot of work to figure out in detail, and this is supposed to be quick and helpful. Second, it's not really true. A product that never adds a new feature is *not* in a "steady-state," because the market is changing around it—customer needs, competitive pressures, the rise of new technologies and trends like AI (p. 419). So, exactly how much new-feature development is needed for "steady-state?" No one knows, and it's not worth your time to try to invent an answer.

Actionable idea arise from components of SSEBITDA.
If SSEBITDA is negative because GPM is low or COC is high, that's an unprofitable model, and you need to address the root causes. If it's especially low due to cancellations or high CAC, those are actionable. If you have intentionally high R&D costs because you're investing in product, and a bump in G&A because you just moved into a new office space, you know those will correct themselves over time. You might even calculate an "expected SSEBITDA after scale" where e.g R&D plus G&A costs total 30% of revenue and see how you're doing.

The overall number is useful as a sort of combined north-star metric, but all the insights and actions come from backing out of the inputs to the number, seeing which of those are worth changing, and acting on those inputs.

Watch it directionally more than absolutely.
This is good advice for most metrics (p. 645). At WP Engine we watched it move month over month from negative to positive and then continue to grow. While you're seeing a positive trend, not just in the overall metric but in the component inputs, and when you have a roadmap designed to continue to improve those metrics, that's a healthy path regardless of the absolute value of the metric today. After all, your goal is not to actually be in a steady state!

Once positive, growing *as a percentage of revenue* could be less important.
Consider that if SSEBITDA is steady 10% of revenue, *and the company is growing*, then in absolute dollars SSEBITDA is growing. Of course it's always great to see improvement in this metric as a percentage of revenue, but certainly it's logical for a company to invest some of it back into the business for de-risking and further growth rather than maniacally increasing this metric.

Now, go get profitable. (Eventually!)

Chapter 92:
Color Wheels are wrong? How color vision actually works

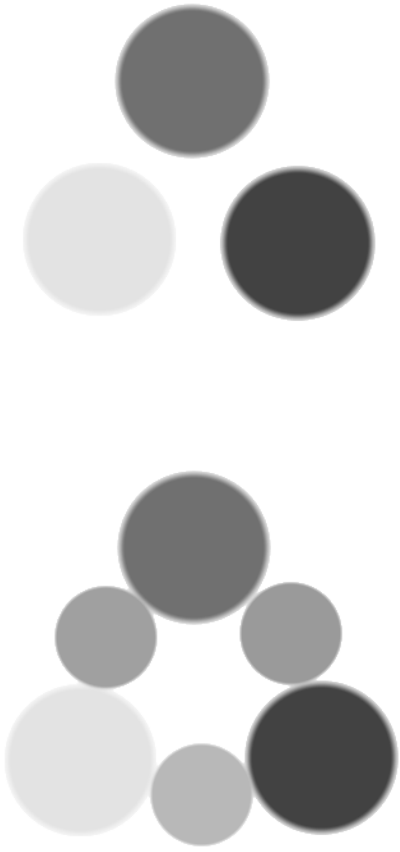
ARTISTS · CONFUSION · PHYSICS · OPPOSITES
PHYSIOLOGY · BETTER WHEEL
BONUS BRAIN BENDER

WHY ARE ARTISTS SPECIAL?

Ask any artist to explain how color works, and they'll launch into a treatise about how the **Three Primary Colors**—red, blue, and yellow—form a color wheel:

Why "wheel?" All other colors are created by mixing these three colors various proportions, they'll explain. In particular, mixing equal quantities of each pair of Primary Colors produces the **Secondary Colors** (orange, green, and purple):

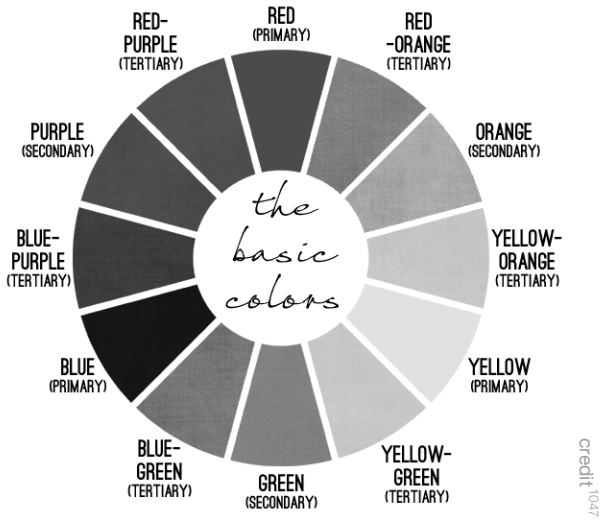
Continuing this process produces the full **color wheel** you might have learned in school; a pretty, symmetrical, satisfying device in which each hue melds seamlessly and linearly into the next:



UNFORTUNATELY, THIS CRUMBLES UNDER EVEN MINOR SCRUTINY

For example, open up your desktop printer and you'll see something quite different:

Three colors of ink which, when combined, produce all others: cyan, magenta, and yellow. (Black is included as a money-saver—black



is the cheapest and most common color; it's cheaper to have a black cartridge than to dump ink from the other three.)

But wait! I thought the "Primary" colors were red, blue, and yellow, not cyan (bluish-green), magenta (bluish-red), and yellow. So one primary is the same (yellow) but the other two are different... yet these still generate color wheels containing all the other colors. **So what does the "Primary" designation really mean?**

Also it's not as simple as saying "*any* three colors can produce all the others" because that's clearly not true (by experiment). And it's not as simple as saying "any three colors will do, they just have to be equally spaced around the color wheel," because yellow is common to both the painter's and printer's wheel, yet the other two primaries differ completely (red and blue are primary in the painter's wheel but secondary in the printer's wheel.)

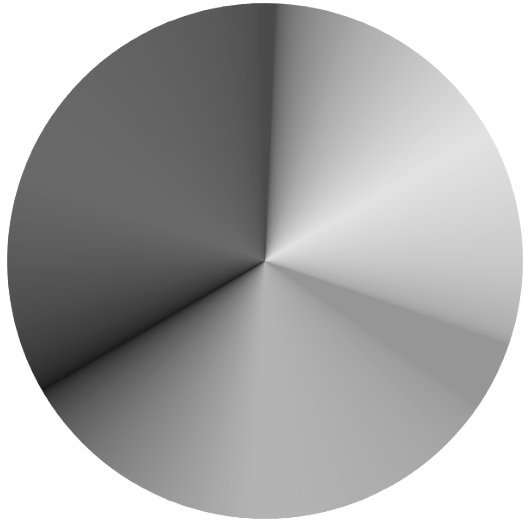
TVs and computers are different yet again. If you stand close to a CRT (non-flat-screen), you can see that every pixel (or "dot") is really three tightly-packed colored phosphors: red, green, and blue.

If you've done computer graphics you've been forced to name colors using these "RGB color values;" true geeks automatically think "yellow" when they see `#FFFF00`. (If it's intuitive to you that `#A33F17` is burnt orange, it's time for you to leave the monastery.)

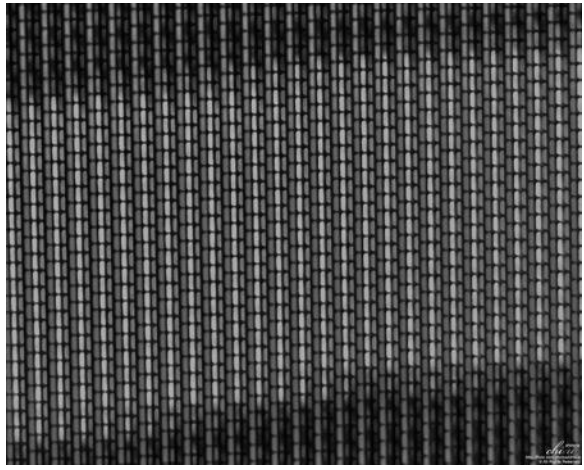
This leads to **yet another system of three "Primary" colors** generating all the others, and yet another color wheel. This one is a little easier to explain—ink and paint are "subtractive" (adding cyan, magenta, and yellow yields black) whereas colored light is "additive" (adding red, green, and blue yields white):

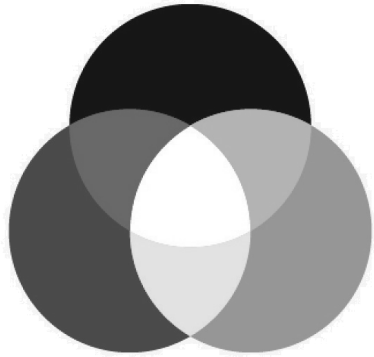
Still, we have yet another color wheel in which two (but not all three!) "primaries" match those of the artist's wheel and none match the printer's wheel.

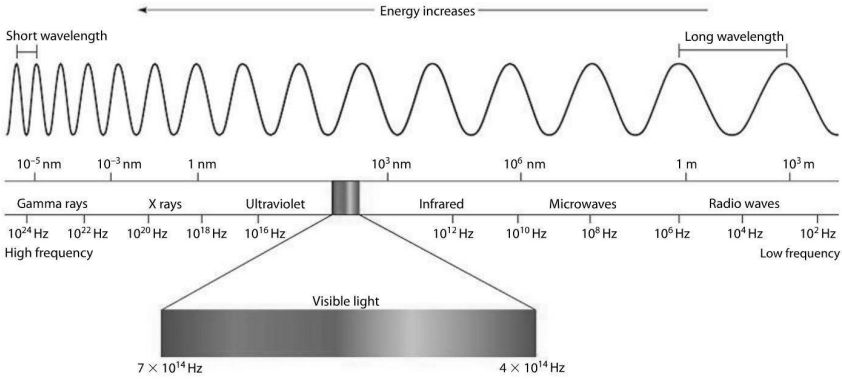
This isn't adding up. **Let's turn to science.**



credit 1048







credit 1049

PHYSICS MAKES IT WORSE

Physics is clear and certain. Light is a wave of electromagnetic energy (and/or a particle, but for today it's just a wave OK?) and, like a vibrating guitar string, light waves wiggle at certain frequencies. Some of those frequencies we detect with our eyes, and the frequency determines its color:

Now we're getting somewhere! Or are we?

First off, we've suddenly lost the notion of a "wheel." As much as the previous color systems have contradicted each other, at least they all agreed that hues transform smoothly and continuously, one to the next, a beautiful symmetry with neither beginning nor end.

But here we have a clear beginning (red) and end (violet). The colors in-between are continuous—and seem to generally match the order seen in the various color wheels—but then it just terminates with violet. How does it get back to red? What about that **fuchsia / magenta / purplish-reddish color** which is **clearly present in every color wheel but missing from the physical spectrum?**

How can a color be *missing*? Where does it come from?

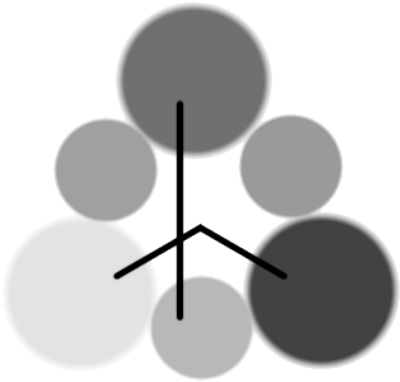
But wait, we're not done being confused.

AND ANOTHER THING: OPPOSITES

Every seven-year-old kid in America is taught that "the opposite of red is green" and "the opposite of blue is yellow." But what does that mean exactly?

After all, there's nothing in that linear physical light spectrum to indicate that any color is "the opposite" of any other, particularly not those two pairs. And the color wheels aren't much help either; trying to match the "opposites" on the painter's wheel yields an unsatisfying asymmetry where two of the primaries are opposite, and the third is opposite from a secondary:

But "opposites" are real. In the early 1800s Goethe (yes, *the* Goethe[1050]) noticed that red/green and blue/yellow were never perceived together, in the sense that no color could be described as a combination of those pairs. No color could be described as "reddish green." If you are asked to imagine "a green with a bit of red," nothing



comes to mind. In the following 150 years, various experiments tested this idea, all of which validated his observation.

There's something to this. Something neither the wheels nor the spectrum can explain.

It's time to get down to the real source of color: The ridiculous complexity of human beings.

THE ANSWER: PHYSIOLOGY (OF COURSE)

Caveat Emptor: The following is a gross and irresponsible over-simplification of what actually happens. But it's correct in its general thrust, and few people on Earth (myself excluded) are qualified to explain with complete accuracy, so in the interest of general illumination, no pun intended, OK maybe intended just a little bit, I'm doing it anyway.

Of course it starts in the eye, where three types of cells called "cones"[1051] measure the amount of red, green, and blue light hitting the retina.

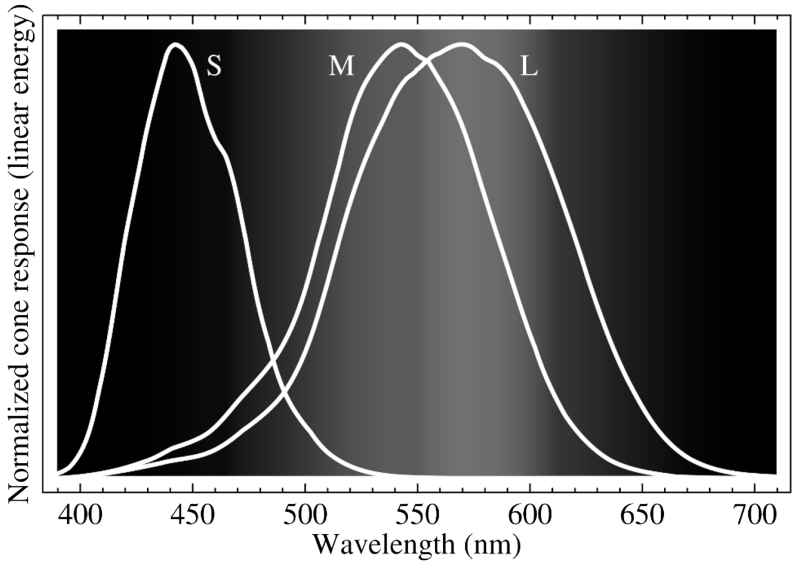
"Ah *ha*," I can hear you CSS freaks scream, "it's RGB after all! I was right! All that time spent—nay *invested*—in remembering that `#001067` is the default title-bar color in Windows 95 was well worth it!"

Hold on there, cowboy. Actually, "amount of red, green, and blue" is a gross simplification (as warned!). Peeking under the hood (just a tad), the three types of cones are in fact denoted S, M, and L for "short, medium, and long" wavelengths, and each respond at different levels in a range of wavelengths:

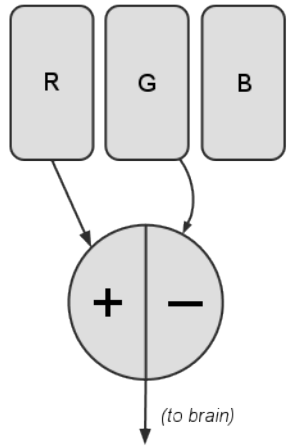
But I digress, and besides I did promise to be all gross and irresponsible, so let's go back to that.

So there are R, G, and B cones. The signals from these cones don't go straight to the brain; they first pass through a pre-processing filter, and **it's this filter that explains all the mysteries.** Actually there are three filters.

Filter #1 works like this:



credit 1052



Explanation: The more R there is, the more positive the signal; the more G, the more negative the signal. If there's relatively *equal amounts* of R and G—whether from none of both, a little of both, or a lot of both—the signal is zero.

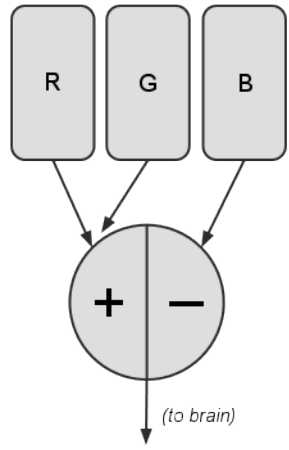
This explains why there's no "greenish-red." Because:

Let's say R and G can go between 0 and 100 units of intensity. Consider the case of "full red with a little green," where R=100 (full intensity) and G=25 (one-quarter intensity). Then separately consider the case of "strong red with no green," where R=75 and G=0.

In both cases, Filter #1 computes the same output signal: 75. But remember the brain doesn't get the raw R and G signals—it only gets the filter's output—so *the brain cannot tell the difference* between these two scenarios.

So there's no such thing as "red with a little green"—there's just a less intense red. The brain physically cannot see "greenish-red" because the filter removes that polarity.

Knowing that blue/yellow is the other opposite pair, you can probably guess what **Filter #2** is:

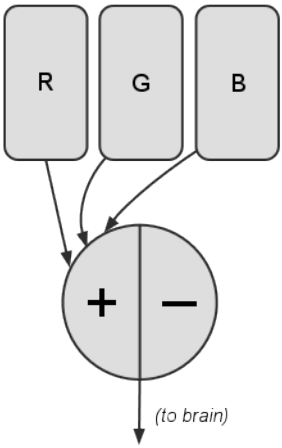


Here blue (B) is opposed with a combination of both the R and G channels. The R and G cones are stimulated either when there's literally both red and green light (like when a CSS coder turns on both red and green as #FFFF00 to create yellow), or when 570nm light (yellow, on the visible spectrum) stimulates both R and G cones.

Filter #3 is simple:

In short, it measures the quantity of light without regard to hue. This is "how bright," or "luminance" in color-theory parlance.

And magenta? It comes from full R and B with no G, activating Filter #1 full-positive, Filter #2 at zero. It's not a physical wavelength of color, it's just a combination of outputs of two filters.



THE PERCEPTUAL COLOR WHEEL

To do this "wheel" thing properly, you should represent the red/green and blue/yellow opposites. It's not at all difficult, so it amazes me how rarely it's seen or taught:

Four primary colors? Yes, why not? It's the closest thing to the physiology without getting complex.

Why is it necessarily a "wheel?" As you trace the (real, physical, see: rainbows) visible light spectrum, filter 1 starts full positive, then goes smoothly through zero and then negative, then back towards one. On the diagram just above, that's the values of the x coordinate of a circle as you trace an angle counter-clockwise starting from pointing rightward along the x axis. So, like cosine, the first filter creates that plot.

Filter 2 does exactly the same, but produces the y coordinate of the circle, like sine: it starts as zero, then moves towards one, then back to zero and then negative, ending towards where it started.

So the color wheel is a simplified, idealized way of plotting filters 1 and 2 through the natural spectrum, and the math of the biological filters naturally plot a circle. Of course the real shape isn't a perfect circle, nor are colors evenly distributed around it, but the general idea is both directionally correct and useful. The CIE color space is closest to perceptual reality:

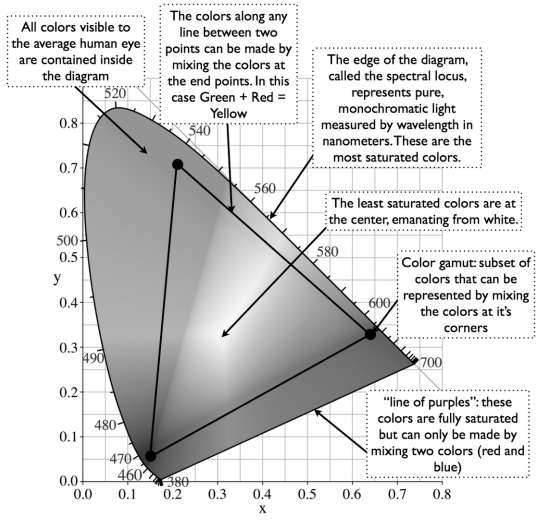
BONUS BRAIN BENDER: THE CONTEXT / COLOR CONNECTION

This is just the beginning of color theory. To give you a glimpse at how complex it gets, consider this:

When a color is juxtaposed to other colors, we perceive it as a different color. For example, most people will say the small square on the left is brown, whereas the one on the right is orange:

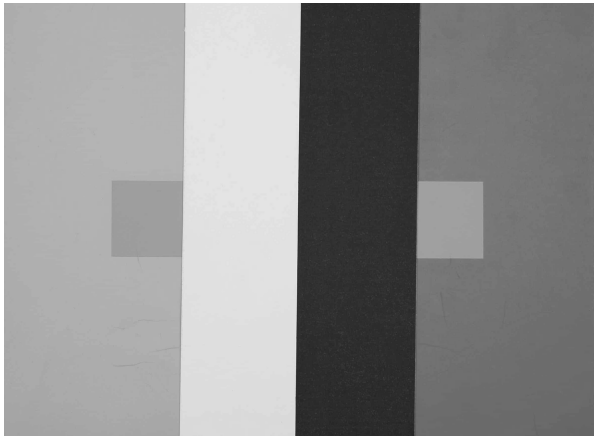
Actually, the squares are *exactly the same color!* The surrounding context dictates the *perceived* color, *on top of* all that wavelength-physiology we just did.

This makes sense because the brain projects abstract things it knows about the natural world onto your perception of color. For ex-



credit 1053

Anatomy of a CIE Chromaticity Diagram



credit 1054

Josef Albers, Folder IV-1

credit 1055

ample, we know intuitively that shadows artificially darken colors, so our brains automatically account for this in our perception of those colors. (It's called "color constancy.") For example, you know that the dark and light colors on this hot air balloon are "the same:"

But it also results in optical illusions so powerful that even when you know the trick you still can't see it correctly.

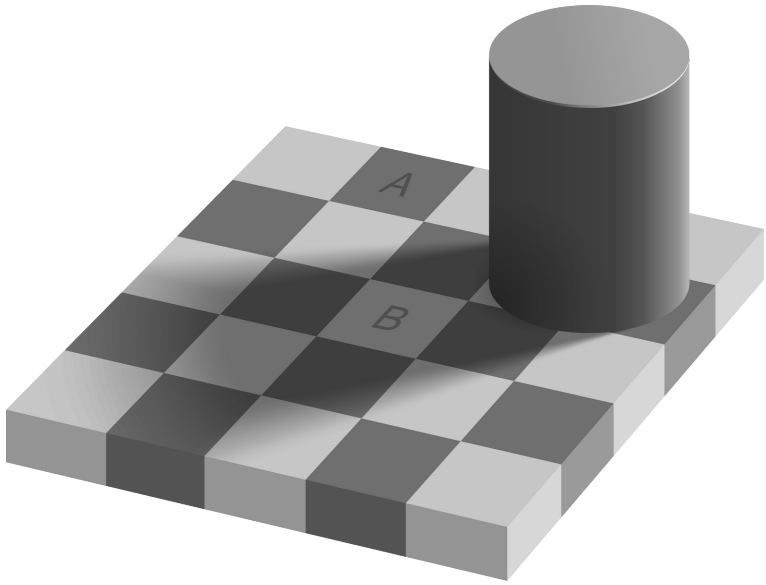
Like this: Which square is darker: A or B?

In fact A and B are the same color (`#787878`), but you can't see it even when you know this. To prove it to myself I had to open this picture in an image editor and actually move one square over another to see it was the same.

Freaky.

Further Reading

You got this far? You still care? Sheesh, you're as weird as me.



credit[1056]

If you really want to lose a few days of your life, this is an amazing, in-depth treatise on color theory.[1057] That link is just page 1 of 8. Good luck.

Chapter 93:
You're a real company when...

Ever since I started Smart Bear in 2002, there's been a running question:

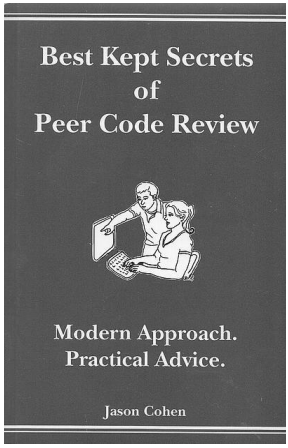
"When do you know you're a real company?"

Which of these are true: You're a real company when...

- You have a domain name that isn't weird.
- Your business cards don't say "VistaPrint" on the back.
- You take PO's in addition to credit cards, and you've gotten over the fact that, yes, you have to actually ship the product before they give you money.
- You switch from Quicken to Quickbooks.
- You hire the first real, full-time employee.
- You pay someone else to print pay stubs and pay your monthly employment taxes because you're just tired of doing it yourself.

- **You have a sign on the building.**
- A real artist does your website and handouts.
- You have a 256-page color glossy hard-backed book[1058] chronicling your ascent in the world (Saw this at Adobe. Gorgeous. I asked everyone I met there about it and no one but the receptionist had ever seen it. It's sitting in the lobby. The receptionist, by the way, knew exactly what their branch of Adobe (Ottawa) did and could articulate why people bought their products in about 15 seconds. Her elevator pitch was better than the 10 minute diatribe I got from a senior manager later that day.)

I made one of those bold because that's the one that's always stuck with me. We got the sign about a year ago, and it really was a proud moment.



Currently Smart Bear has everything but the glossy book (but we do have a book[1059] with a glossy cover with over 7000 copies in circulation). The 8000'th user of our various software products just came online. That makes me feel like a real company. *[2009 Update: now 45,000 books and 30,000 paying users]*

That was written 16 years ago. Now in 2023, I have a new story.

I needed a new one... it's not hard to pay employees anymore, and you've probably never heard of VistaPrint, nor used Quicken. Well, it used to be funny (Figure 1).

Ben Radler says:
January 19, 2010 at 03:09 Edit

Your business cards don't say "VistaPrint" on the back.

This made me laugh out loud, for serious.

David Cruz says:
April 10, 2010 at 10:37 Edit

"You have to switch from Quicken to Quickbooks."
That made me smile :)
.-= David Cruz's latest blog post: Soon, a PC will not have files, folders, or documents ↵ =-.

Figure 1

When I was a little boy growing up in Austin Texas, "Downtown" was this frightening place we never went to. Downtown had full of things like government buildings and law firms and things that families didn't go to, and there was crime.*

But often we would come to the outskirts of Downtown to get bagels. My parents grew up in and around New York City, and they insisted there was only one place to get "real" bagels. So sometimes, on the weekend, we'd get in the station wagon, I would hop in my car seat, and we'd go down to the bagel store on the edge of Downtown. Looking out window, I knew we were at the bagel store when I could see a long, sloping ramp that led a parking garage. I would wait in the car, staring at that scary office building across the street, they'd get the bagels, and then we'd immediately U-turn and get out of there.

Fast forward 40 years, and I'm standing in a public park with our amazing CEO Heather Brunner and a few dozen of the many hundreds of people who worked at WP Engine, watching as our sign goes up on the building. We kept checking a picture of the company logo on

* Austin was 1/10th its current size, and probably didn't have that much crime, but to a little-boy brain, it was scary.

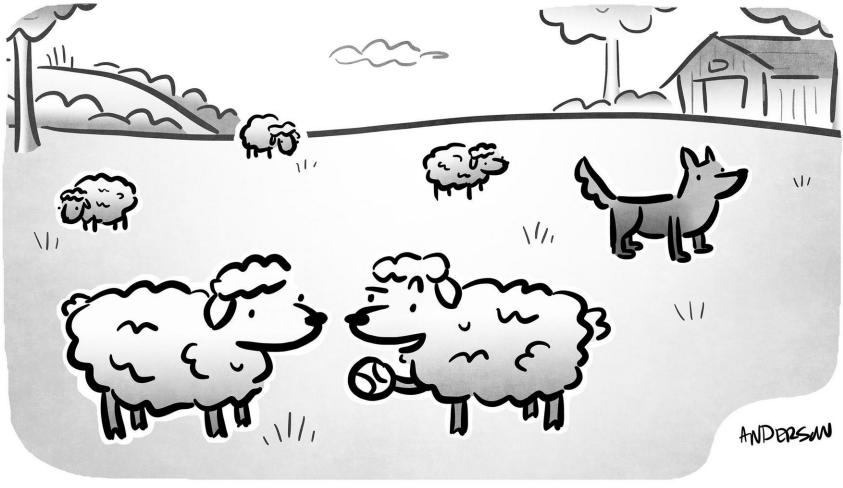
our cell phones, ensuring that each little piece was correctly oriented. (We couldn't remember ourselves!)

That building, is the building with the sloping ramp, imprinted on my memory like a duckling on its mother, one of those inexplicable snatches of childhood memories.

You're a real company when...


WP engine

Chapter 94:
Creating space



"With this humble tennis ball, Phil, we shall forever change the status quo."

credit 1060

The one-word advice given to all startups, and to CEOs of large companies, and they in turn to their teams:

"Focus!"

But what does that mean, precisely? And why is it important; isn't quantity sometimes in fact better than quality?

Because only when we stop doing most things, do we have the time and energy to fully, deeply execute the most important things. And the important things are the difference between thriving and perishing.

When you stop trying to sell people who aren't a good fit (p. 479), you create space for 3x more sales pitches, each with 3x the close-rate.

When you help draining customers exit the business,[1061] you create space to spend time with 3x more easy, happy ones, who want to spend more every year while giving you positive energy.

When you stop trying to win over everyone, you create space to win over the right ones (p. 317).

When you stop trying to shore up every weakness, you create space to leverage (p. 543) your strengths.

When you stop trying to schedule all work, you create space for impactful work (p. 221).

When you stop trying to do everything in a novel way (UX, coding, pricing, org structure), you create space for a few truly impactful ways the product or culture is differentiated.

When you stop using lazy, generic words in marketing, you create space on the page to communicate what you actually do (p. 1439), or who you actually are (p. 627), or why anyone should care (p. 1169).

When you end toxic relationships, you create space for healthy ones. Personal or professional. For you and the teams around you.

When you stop trying to control every last detail, you create space for empowered teams to flourish (p. 413), and for you to work on things that only you can, or only you ought.

When you stop reading garbage, you create space to read something meaningful, or useful, or enjoyable, or inspiring, or refreshing.

When you stop chasing every metric, you create space to attack the one metric that will transform the company (p. 1125).

When you stop trying to improve everything by 1%, you create space to improve the biggest thing by 30% (p. 925).

When you stop trying to please every ill-fitting customer, you create space to convert "satisfied" customers into "fanatics (p. 1559)" who increase word-of-mouth growth, leave positive reviews, and never cancel even if something bad happens.

When you stop checking email and social media every 11 minutes, you create space to become enveloped in the flow of creativity and productivity.

When you stop trying to maximize every metric (p. 887), you create space to maximize the few metrics that matter most (p. 645).

When you stop pursuing every opportunity, you create space to fully capitalize on the one with the thickest intersection (p. 867) of upside potential and ability to execute.

When you stop creating superficial relationships with every stranger on social media, you create space to create deep relationships with people who care deeply for each other.

When you stop trying to win every battle with every competitor, you create space to win the ones that matter most to the people in your ideal market segment.

When you stop trying to "find the balance" in everything, you create space to maximize the one choice (p. 589) that creates clarity and strength.

When you stop trying to "have it all (p. 901)," you create the space to deeply experience the few things that are most important.

We don't have unlimited time or energy, but we can better spend the limited time that we have (p. 725).

That's why: *Focus.*

> Decisions are easy when you have only one priority.
> Your destination is a huge mountain peak on the horizon.
> You can see it from everywhere.
> Yes to that mountain, and no to everything else.
> You'll always know where you're going, and what you're doing

next.
All paths go either towards that mountain or away from it.

Because of this perspective, problems won't deter you.
Most people look down at the ground, upset by every obstacle.
With your eyes on the horizon, you'll step over obstacles,
undeterred.
—Derek Sivers, *Hell Yeah or No*

Chapter 95:
Leverage Points

TOP OF FUNNEL · PRICING · CANCELLATIONS
ONBOARDING · WEBSITE · HONORABLE MENTIONS

"So you believe that when you flap your wings on one side of the planet, all kinds of crazy things are caused on the other side? That sounds like a lot of responsibility."

credit 1062

Every business has a few special areas where small changes in input have large changes in output. It is efficient to spend time and money in those places, winning an out-sized return.

If only you could identify what those Leverage Points are!

Ideally, everything important would be a number (p. 645), precisely reported in real-time. Ideally, you could analyze flows like attraction → consideration → purchase → onboarding → active customer → cancellation. Ideally, you could run 100 tests simultaneously, with enough data points to identify subtle changes, repeated to eliminate the inevitable false-positives (p. 913). Ideally, you could experiment in one area without affecting the data in another. Ideally, but sadly, rarely.

Fortunately, I have found that most companies share a few common points of leverage. They are common because they exhibit some natural mathematical leverage that applies to many business models.

Therefore, if you're part of the non-ideal majority, you can be confident that focusing on one or more of the below will have a large effect on your business.

Pick one?

TOP OF FUNNEL

What if your marketing were so perfect that only the only people who clicked your ads were ideal customers (p. 317) from your target market (p. 1211)? People who already feel the pain you solve (p. 71), identify with the way you talk about it (p. 165), want the specific set features you currently offer, have an allocated budget that exceeds your price (p. 1223), and are wanting to purchase today (p. 479).

In this scenario, even a mediocre website, an imperfect conversion funnel, and a talentless order-taking sales team would convert most of these folks to a sale, simply because they're the right people. This al-

ready demonstrates that the top of the funnel is a Leverage Point: The system works well, even when so much else goes wrong.

Now consider what actually happens in many companies. The direct-marketing group is tasked with "increasing traffic." So they pay for traffic that turns out to be lower quality than usual. You might think that, while this is a less-efficient use of money, it's fine, because *some* of that new traffic will become customers, and the rest will get filtered out by the rest of the funnel. But reality is far worse.

Lower-quality traffic decreases website-conversion rates. The website-conversion team now has trouble measuring their long-term success at improving their metrics. While you might think that a single A/B test won't be affected (because both sides of the test experience the same lower-quality traffic), in fact it *is* affects, because finding "winning" variants will require appealing to low-quality leads, rather than appealing to the ideal lead who will actually become a customer. So this means they're doing a poorer job converting the ideal customers, and a better job passing the wrong customers through to the next step of the funnel, where either sales teams waste time in pointless sales calls, or short-term cancellation increases as customers figure out for themselves that the product was never a good fit. All of it wasting customers' time, wasting employees' time, wasting company's money.

"Optimizing" one step in the funnel for quantity or conversion rate, rather than for quality, affects all downstream steps. Therefore, small changes in the quality of the first step in the funnel has a dramatic, out-sized impact on the effectiveness of the entire company. This is leverage.

On top of that, it breaks our ability to learn and improve with the right customers. The web team can't optimize for the right customers, because their tests are polluted with the data from wrong ones. The

product team can't tell which feature requests come from good customers as opposed to people who should never have been customers, pulling them in the wrong direction. Tech support spends time with hopeless cases instead of thrilling customers who want to be here for the next ten years, and can't analyze what topics are worth addressing in the product. Finance can't report the metrics we need—CAC, ARPU, NRR, cancellation—because the true numbers are mixed with garbage numbers. Sins at the top of the funnel infect everything else.

Marketing funnels aren't the only systems that exhibit this effect. Consider a company who says they "only hire the top 1%." Not true. At best, they hire the top 1% *from the resumes they saw*. Surely, many of the actual "top 1%" people never applied. And just as surely, the definition of "top" is unclear, and the ability of an interview process to correctly discern which people are "top 1%" is dubious. So, what is the truth?

The truth is, if you're sourcing from a pile of fantastic applicants, then your interview process and your talent discernment doesn't matter. You'll hire good people regardless, even by accident, even if you pick at random. Whereas, if your applicant pool is weak, than even the best discernment process in the world will still not result in strong hires. The best people were never there.

Or consider angel investors, who talk to companies that are seeking funding, typically before the company has much evidence that it will work. The angel tries to select the "best bets" from the set of companies they see. "Best bets" is in quotes, because angels are famously terrible at picking winners. The median angel loses money,* and even the top angels in the world still have more failures than successes. An investor who sees nothing but stellar opportunities will build a decent portfolio almost regardless of which companies they back. Whereas, an investor with bad "deal-flow" (as it's called) will not succeed, even if they pick the best companies from what they see.**

---

* Even among professional venture capitalists, 65% lose money,[1063] and only 10%[1064] generate returns high enough to justify the risk and illiquidity.

So, "top-of-funnel-quality" is a Leverage Point in any funnel-like systems, which applies to any such process in any company.

PRICING

You already know price is important (p. 1223), but you might not realize why it is nearly always true that small changes in prices—in either direction—have an outsized impact on the business.

In microeconomics you learn the relationship between price and sales: If you increase price, you decrease the number of people "willing to buy (p. 275)," but each one is giving you more money. So the question is: Is the increase in revenue from higher prices big enough to compensate for the fewer sales? If yes, the economist would say your prices are "too low," else they are "too high," and ideally you hone in on this sweet-spot maximum amount of revenue (Figure 1).

In reality, this is rarely how it works. (Economic theory often fails to predict (p. 193) the real world.) In reality, even small price changes sometimes cause sales to drastically fall off a cliff or, especially in early-stage startups, increases in price counterintuitively lead to an *increase* in the number of signups (Figure 2).

Why? Because price isn't independent from the product or the target market; indeed, price is a main determinant of your target market

** This also explains the persistence of top venture firms. Once a firm establishes a stellar reputation, the best founders will proactively seek them out, resulting in great portfolios almost regardless of their ability to pick. This is not undeserved; successful firms also attract great people to work there, and by all accounts really do help companies get to the next level. Nevertheless, their continued good results is made possible by (earned) deal-flow more than their ability to select individual winners. More evidence of this: Looking backwards from successful IPOs, software companies are typically backed by one of a few well-known firms, yet other well-known firms passed on those same deals. It didn't matter which firm said "yes."

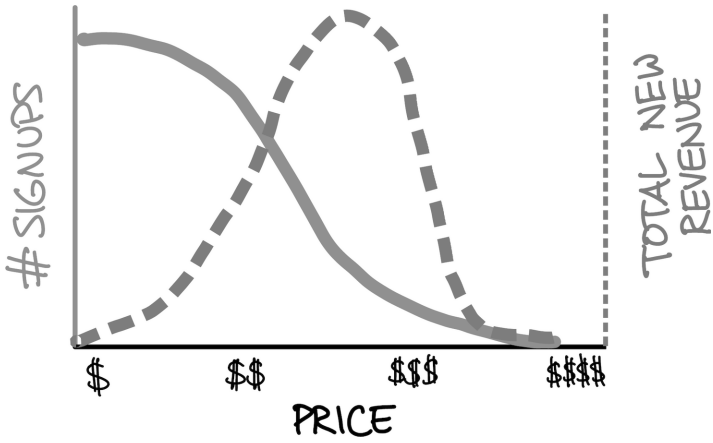


Figure 1: As price increases, fewer people buy. Total revenue is signups multiplied by price, creating a natural maximum that economists insist we seek.

(p. 1223) and your business model (p. 515). Therefore, changing price can mean changing the target market (p. 1211), which changes everything.

Clearly, a large change in pricing represents a large change in strategy, positioning, target market, and customer expectations. But the question here is whether *small* changes also have large outcomes, contrary to micro-economic theory.

There are three possible results from a small price increase:

(1) Has a measurable (i.e. large)* negative effect on new customers per week.

You are in a highly competitive market without significant, valuable product differentiation. Because if it weren't competitive, customers wouldn't have an alternative, and if your product were differentiated, they couldn't select an alternative and still get what they want. There-

* Because statistical-significance is hard (p. 913), especially with relatively low N that even established companies might have in weekly signups, here the word "measurable" also implies "large." And by assumption the change in price was small.

Figure 2

fore, there are two actionable conclusions:

(a) Lower your prices. You've demonstrated that the micro-economic model is correct in your case, so you can not only get more revenue from lower prices, you also get more customers. "More customers" is always better, all else being equal (though "all else" is never, in fact, equal). It means more people recommending you, more product feedback and insights, more momentum, lower impact when one customer leaves, and fewer customers for competitors. Warning: Don't lower prices so much that you change your target market, unless you're intentionally making a strategic change. Blindly competing on price is dumb, but strategically competing on price can be brilliant (p. 437).

(b) Create differentiation. There are many possibilities.* Perhaps your

product is, in fact, unique, but you're not effectively communicating that (p. 949). Maybe your product is unique in aspects that customers don't value. Maybe you need other advantages of Love and Utility (p. 275), like having a higher purpose (p. 1169) that customers want to support. Most likely, you need to face the truth (p. 657) that your product really isn't special in a significant way.

(2) Has no measurable effect on new customers per week.
In this case, slightly raising prices is a pure win—more revenue, more profit. The reason this is also an outsized result, is that it increases profit non-linearly. Suppose a business charges $50/mo and nets $5/mo per customer after all expenses. Raising prices even just to $55/mo nets $10/mo. That is: a 10% increase in price resulted in a 100% increase in net profit. That is the definition of leverage.

Your next task is to raise prices a little more. Raise until something changes—you're no longer targeting the right customer, signups fall off, or some other clear signal that you have in fact found your optimal price.

(3) Has a measurable (large) positive effect on new customers per week.
As shown earlier, this can happen, and even the possibility of it is reason enough to try. Obviously you should keep that pricing, and like the previous case, your next step is to keep trying raises until something significant changes.

It is quite likely (though not certain) that this also indicates you were incorrect about your target market. Customers apparently have budget, urgency, and pain-points that are significantly different than you thought. The reason the numbers are moving contrary to economic theory is because you're tapping a new customer segment, carrying different dynamics. Great news, but you need to dig into "why" just as fervently as if the result were negative, so you can learn what your customers are really like, what your target market really is, and then how to update your strategy and product. And congratulations, you're probably at Product/Market Fit (p. 335).

---

* This great article by Ton Dobbe[1066] details many of these possibilities, how to identify them, and what to do about it.

No matter the result, you have to find out.

CANCELLATIONS

It's almost impossible to get a customer.

They had to hear about you, whether through the white noise of social media or the labyrinth of organic search results, or desperate advertisement, all of which has never been more saturated and engineered. They had to click through to your website. They didn't bounce off in 3 seconds. They actually read something. They understood what you offer. They decided it might work for them. They were unfazed by the pricing page. They didn't immediately reject you after comparing with competitors. They signed up. They entered a credit card number. They on-boarded themselves, configuring things, entering data, finding early success. They invested hours. They engaged with support. They really *wanted* it to work.

And then, after all that, they cancelled.

Even without the financial argument, this is already enough to know that understanding cancellations must be one of the most critical things you can do. If the customer was willing to do all that, and still your product fell short, you *have* to understand why.

Maybe you promised something you're not delivering on. Maybe you did deliver but the product wasn't intuitive enough for them to figure that out. Maybe the product is useful on day one but not day one hundred. Maybe the target market can't really afford it or isn't really serious. Maybe you're asking the customer to make changes they can't make. Maybe your competitor is better at product and proactive marketing. Maybe bugs or gaps are pushing them away. Whatever it is, you *have* to find out, and make changes.

And then there's the financial argument, which proves that it's not just critical for product, it's a Leverage Point, because small changes in cancellation rate result in large changes in revenue. The reason is simple: Cancellation is exponential, but nothing else is; specifically, marketing is not (p. 115). Cancellation is exponential because it's proportional to the size of the company, which is why it is measured as a percentage, e.g. "7% per month." Marketing is measured as leads per month and ROI per leads. The bigger you are, the greater the effect of cancellations (in absolute dollars), but running ads works the same regardless of your revenue or customer-count.

Making this concrete: With $15k MRR, adding $2k/mo of new customers—a healthy 15% *per month* growth rate—a 7% cancellation rate means already half of that growth is negated by customers leaving. The company barely got started and already its growth is being decimated. At that rate, only one year later, having grown to about $27k MRR, the company has stopped growing completely (Figure 3), despite spending time and money on marketing and sales.* Don't forget—those new customers cost money to attract, sell, and on-board with tech support, but all the value of that expense is negated by an equal number of customers walking out the door.

As described in detail in this article on how to calculate your maximum MRR (p. 1153), changing your cancellation rate has a large impact on growth rate and how large you can ever grow. In our example, moving from 7% to 5%/mo increases our maximum MRR by 50%, to $44k. That would enable use to hire two people, or spend more on marketing to increase that $2k/mo. It would dramatically transform the company's capabilities.

Whether you're swayed by the math or the product or the thought of being rejected by customers, lowering cancellations is almost always a high-ROI activity.

---

* Growth stops because the $2k of new customers arriving in a month are negated by $27k × 7% ≈ $2k customers cancelling in that same month.

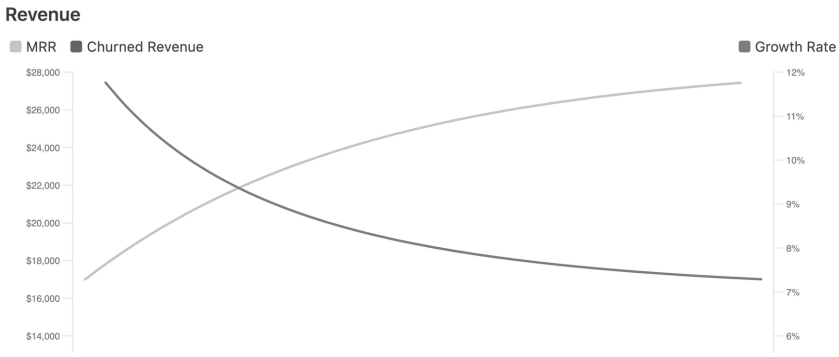


Figure 3: Charting adding $2k/mo new MRR with 7%/mo.

ONBOARDING OPTIMIZATION

Starting with the chain of events from the previous section, stopping at the "onboarding" phase, we find another common Leverage Point.

A common mistake in analyzing customer retention is to look at total retention across all customers. Instead, look at retention based on the age of each customer, by month or even by week. You will find a large drop-off in the first 2 or even 10 weeks, followed by relative stability. This is the difference between customers who successfully onboarded and customers who have settled in. When you see the stark difference, you realize they are totally different beasts, which means you need different metrics and different actions.

It's not just SaaS products. Consider this chart of how many people are still watching my YouTube video about the Profit Whale Curve [1067] (Figure 4). 20% of viewers made it past 20 minutes—pretty good for brain-rot YouTube viewers. But 50% dropped off after merely 30 seconds! If you watch, you'll notice that my hook isn't great, and then I waste time with an indulgent joke about constellations—not cute enough to hold people's attention. I can improve.

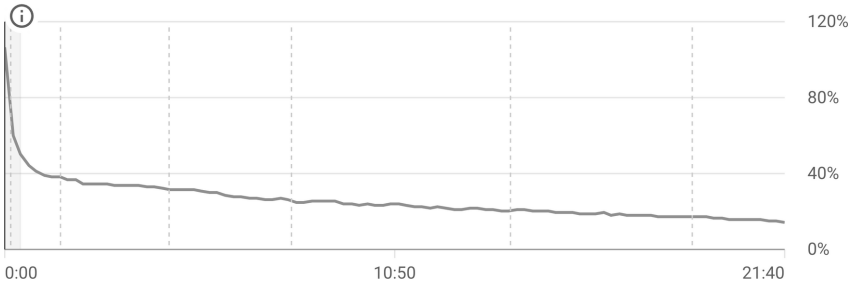


Figure 4: Only 50% of viewers are still watching the video after 30 seconds. Another 50% survive the full 22 minutes.

If I shift that number from 50% to 55% (10% improvement), it's possible that viewers-until-the-end could shift from 20% to 25% (25% improvement). Leverage, and a better video.

This also applies to SaaS products. Early in Hubspot's life, they realized that new customers who watched their on-boarding instructional videos were far more likely to still be customers after six months. So, they started enforcing an onboarding training hour as part of their sales process, shifting[1068] their monthly churn rate from 3.5% to an incredible 1.5%. Hubspot investor David Skok witnessed[1069] similar transformations at other companies. Better on-boarding also[1070] decreases total support contacts and increased customer happiness.

Onboarding is also when customers are most engaged and receptive. They've just committed to your solution and are motivated to make it work. This psychological opening closes quickly—within days, not months.

There are too many benefits to ignore. Onboarding is a Leverage Point.

WEBSITE OPTIMIZATION

Every marketing channel you invest in—whether it's PR, social media, paid ads, or word of mouth—funnels through your website. Therefore, any improvement to your website's conversion rate multiplies across all your acquisition efforts. A natural Leverage Point.

Beware the common claim that multiple improvements compound. The usual argument goes like this: A 20% improvement in your home page headline, a 15% lift from a better "Try Now" button, and a 25% increase from a clearer pricing page doesn't add up to 60%—it multiplies to a 75% overall improvement in conversion rate. While this sounds logical, the real world doesn't work like this. The reason was given earlier: When you improve one point, you affect other points, because you've changed who arrives there.

Another mistake in the typical advice is that small aesthetic changes can have big results. Sometimes they can, but large changes are more likely to have large results. Changes in positioning, attitude, language, consistency, and alignment with the target customer. This video shows an interesting real-world example (Figure 5).

However, even a 20% improvement in website conversion yields a 20% increase in growth rate and a 20% decrease in the cost to acquire a customer (p. 1377). This double-win makes even small improvements extremely valuable; the definition of leverage.

To earn that 20% improvement, create theories about what customers from certain inbound sources are thinking, what they want to see, how they want to see it, what they want to do next, and then use A/B tests to disprove or hone those theories. This article explains how (p. 913), and also why you must do this rather than the usual "throwing shit at the wall and seeing what sticks."

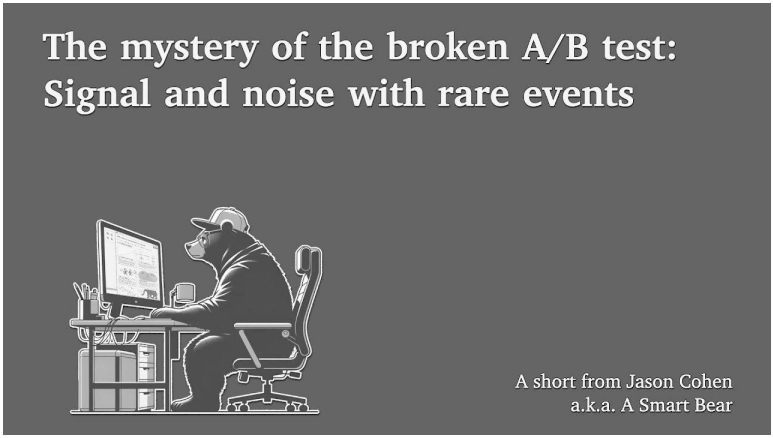


Figure 5: Watch on YouTube[1071]

HONORABLE MENTIONS

There are more areas that have enormous impact for most companies, but that typically require more than just a small change. While these might not completely qualify as a Leverage Point, they are worth your consideration:

Prioritization
Different from productivity; if you run fast in the wrong direction, it's still wrong. Therefore, it is enormously impactful to improve your ability to collect ideas (p. 711), prioritize different work differently (p. 221), and focus (p. 1119) on the most impactful things (p. 833).

NRR
Net Revenue Retention (NRR) refers to the growth of existing customers —upgrades, downgrades, and cancels, but not new customers. When it is positive, it means the company is growing even without marketing; this dramatically changes the growth curve, as much as cancellation alone. The reason it's here, is that often making changes here are not easy or small, as it involves significant new tiers or new products.

Pricing terms
There's the price, and then there's how you charge. Annual pre-pay can transform (p. 353) the cash-flow of the business. Understanding budget limits can be the difference between a customer believing the product is actually $0, or conversely can be so annoying that they just buy a competitor instead.* Buying out a competitor's locked-in contract can win larger customers. Payment plans can win smaller customers. Including services which are relatively inexpensive for you can win deals. Trading a lower price for strong public testimonials or beta-testing can pay for itself a hundred times over.

Channel power laws
Many acquisition channels follow a power-law where the top 1% yields 20% or more of the results. The top Google ad or the top organic search result gets twice the clicks of the next. Top affiliates generate more customers than all other affiliates combined. In these cases, getting a top result has leverage, but playing in the long tail might not be worth your time at all.

Team
The best teammates increase everyone's productivity, improve morale, contribute to culture, and make work fun, even when you're tackling existential crises. That goes for management (p. 413) too.

Purpose
Having a purpose (p. 1169)—whether as a result of execution or as the *raison d'être* for the company—motivates both employees and customers, keeps customers who otherwise would have cancelled, aligns strategy, and improves morale.

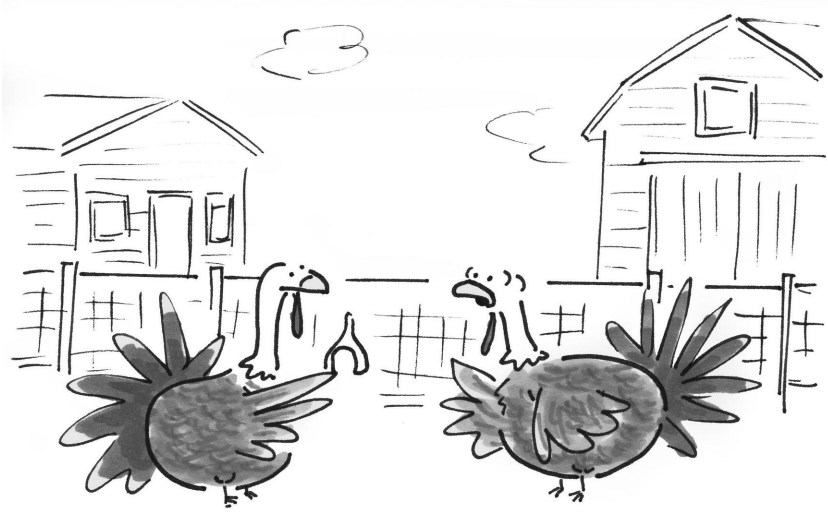
The general idea is simple: Your time is limited (p. 725), so you must focus (p. 1119) on high-leverage activities. Details will of course

* This happened to me; here's that story (p. 1239).

depend on circumstances, but if you tackle one of the themes in this article, it will almost certainly be a good use of that time.

Chapter 96:
Lost confidence

CONFIDENCE GAMES · TECHNIQUES



credit[1072]

CONFIDENCE GAMES

Many prioritization frameworks include a measure of confidence*—how sure we are that we can execute, at more-or-less the predicted estimated effort, resulting in more-or-less the predicted impact. This seems rational; if two projects generate equal value for equal effort, but we're confident we can execute the first and unsure about the second, we should select the first.

This is not, however, how confidence scores are used. If it were, the process would look like this:

1. Score projects somehow.
2. If there's one clear winner, do it.
3. If there's a tie, pick the one we are more confident in.

That's not a bad idea. But popular frameworks like RICE include "confidence" in step one:

$$\text{Score} = \frac{\text{Reach} \times \text{Impact} \times \text{Confidence}}{\text{Effort}}$$

Or RPS:

$$\text{Score} = \text{Reach} \times \text{Potential} \times \text{Solution Confidence}$$

Which means, for example, the following two scenarios are deemed equally strong:

---

* Or a measure of risk. Whether `risk` is equal to `1-confidence` is left to the discretion of the reader.

1. A small incremental feature, that we're sure we can execute.
2. A large feature, with large impact, that carries some risk.

This equality is false. Especially when you remember that small projects almost always carry higher confidence, and rightly so.* But that systematically skews the prioritization away from delivering as much value as possible—the opposite of what a prioritization framework ought to do.

I don't believe your confidence score anyway. First, because it's ill-defined. What does "30%" mean? What it *should* mean (p. 997), is you track your confidence scores and measure how accurate they were after the fact, and determine how good you are at it with mathematical precision (p. 1319). But you don't do that, do you? And if you only ship a few major features per year, you don't have enough data to know.

Second, I don't believe you because we all know that projects are nearly always late, and often have less impact, less quickly, than we wanted. No matter how confident we were. Indeed, *everything* we choose to do, we have at least "pretty good confidence in," or we wouldn't do it at all! So what weight should we place in "confidence?"

> **Hofstadter's Law**
> It always takes longer than you expect, even when you take Hofstadter's Law into account.

To prove this, find any experienced Product Manager (p. 817) and ask: "Can you recall a feature you were certain would be well-received, but wasn't?" Perhaps they had evidence from customer conversations, explicit requests, or purchase commitments—yet after building it, almost no one used it, including those who promised they would. Their eyes will roll as they share multiple stories. This doesn't make them a

* If you disagree, consider that the entire motivation of the Agile movement was that we should *always* have low confidence that large projects will be successful, despite our best techniques of planning, analysis, and estimation. And consider this theory of Rocks, Pebbles, and Sand (p. 221).

bad PM. Everyone who has built products, regardless of skill, has these experiences. The best PMs have techniques to mitigate this problem,* but none will claim they can eliminate it entirely.

Similarly, ask content creators about their most successful work. Often it's something they hastily produced—a trivial piece they almost didn't publish because it seemed uninsightful or trite—yet it generated more views and engagement than anything else that year. Conversely, pieces they spent dozens of hours crafting, work they're genuinely proud of and consider their best, generate minimal interest (Figure 1).

We can summarize the relationship between our confidence and actual results in a handy two-by-two table:

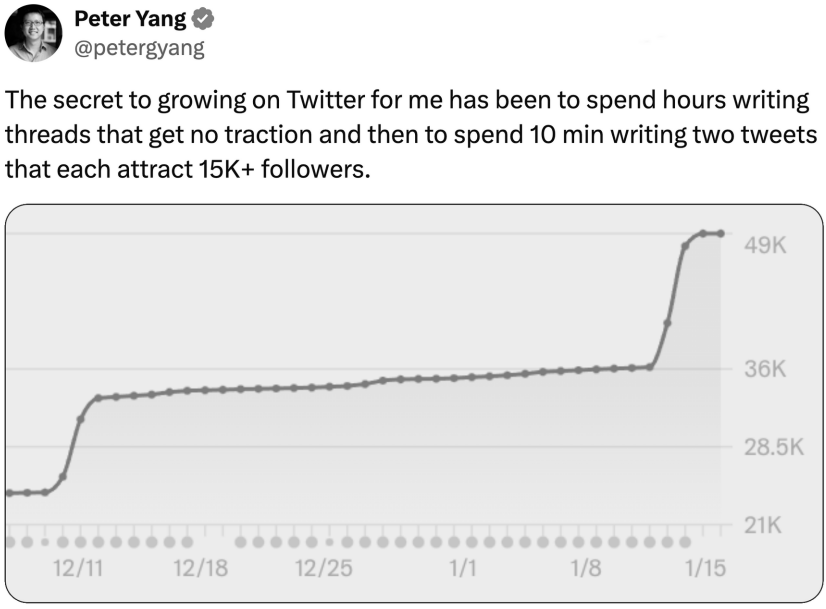
	Was confident	**Was not confident**
They loved it	Lots of things	Lots of things
Nobody cared	Lots of things	Lots of things

So, if "confidence" is too nebulous to define, and we shouldn't trust ourselves with it anyway, what should we do?

WHAT TO USE IN PLACE OF CONFIDENCE AND RISK

The answer lies in the realm of uncertainty, rather than of probability.

* Some techniques to improve prediction include asking customers to describe exactly how they would use a feature in their normal workflow. Often people genuinely think they would use something, but when forced to walk through it step-by-step, they realize, "Oh wait, this would require me to rewrite this code, we probably wouldn't do that." Or, "I'd need to export it into another system—actually, never mind."



credit [1073]

Figure 1

Probability presumes you know the underlying distribution, enabling mathematical predictions about future events. You can predict that flipping a fair coin 100 times is highly likely to result in between 40 and 60 heads, because you know the underlying distribution. If predicting whether a feature will create defensible differentiation were like coin-flipping, you could use probabilities.*

Almost nothing in a startup is like that. Outcomes cannot be assigned meaningful probabilities because things like startup success, strategy, and features are unprecedented, or too complex to model accurately, or we have no precision on the input variables. This is the domain of uncertainty.**

* If you're tempted to claim that Bayesian methods could still work, remember that you need numeric priors and conditional probabilities, both of which we established above are unknowable and ill-defined.

In this domain, we ask: What actions are wise regardless of the probability distribution?

I've previously written about embracing uncertainty in overall product strategy (p. 193). Below, I'll address a more specific question: How should we prioritize individual items in an uncertain world?

Here are some techniques.

True always

What is always true under any circumstance? This is Bezos's principle of focusing on long-term constants.*** For instance, users universally appreciate fast, responsive software. They value web apps that feel native, with background synchronization and instant interactions, that work well on all their devices. At worst, they might not consciously notice; at best (in web-apps like Notion, Miro, Gmail, and Google Docs), performance becomes a key differentiator that customers explicitly value.

Not all features enjoy universal appeal. Rather than attempting precise numerical breakdowns of potential user interest, identify the features where essentially all customers will either value it, or at least enjoy it. Sometimes this certainty exists because is mandatory, even if mundane. Enterprise requirements like SOC 2 compliance aren't exciting, but they're undeniably valuable when selling to the Enterprise. This certainty compensates for the lack of differentiation.

The caveat: your most innovative, differentiated ideas rarely fall into this "absolutely certain" category. While certainties are valuable, they're unlikely to be your strategic differentiators. This tension is nat-

** Formally called "Knightian Uncertainty" after economist Frank Knight in his 1921 work "Risk, Uncertainty, and Profit."

*** Bezos frequently said of Amazon's strategy: When you have something that you know is true, even over the long term, you can afford to put a lot of energy into it. His examples include customers wanting lower prices, faster shipping, and fast, fair customer service.

"Absolutely! 100 percent! Guaranteed!
Most of the time."

credit[1074]

ural—great products require both reliable improvements and innovative leaps of faith.

Quick discovery

I've been a long-time advocate of systematically interviewing potential customers (p. 239) to validate ideas before you start building. Still, I have to admit that this falls into the "confidence" trap. You never really know until you build. (You can, however, invalidate before you build, saving you months if not years of wasted time; therefore this is still the right place to begin.)

The typical solution is to build an SLC (p. 101) (my upgrade to an MVP), i.e. a completed but simple product that generates real feedback. Experience, rather than prediction. For existing products, that means maintaining a balanced (p. 589) portfolio between guaranteed wins and innovative bets, applying different validation methods to each.

For example, consider implementing "dummy features"—buttons that, when clicked, reveal: "This feature isn't built yet. Tell us how you'd use it." This simple test provides real signals: a count of interested users and potential interview candidates who've demonstrated interest through action rather than words. They can provide insights before you build the feature.

This approach generates 100x better signal than surveys asking hypothetical questions. People easily say "yes" to survey questions about future usage, but taking an action—even clicking a button—requires genuine interest. Observed behavior beats stated intentions every time.

Customer impact

Replace confidence with impact. I define impact in two distinct ways:

Majority rule
When the majority of users regularly use a feature, it's undeniably important—likely a key reason people adopt and retain your software.

Passionate advocates
Features that create passionate advocates among a smaller subset of users. These "magnificent delighters" won't appeal universally, but they inspire deep loyalty in specific segments. Like a piece of music that's someone's all-time favorite (while others merely acknowledge it's objectively good).

These are what determine purchase decisions. Your product rarely satisfies every customer need perfectly, but when users absolutely love certain aspects, they'll tolerate shortcomings elsewhere. We see this with beautifully designed (p. 853) software—users accept missing functionality or limited platform support because the design experience itself is so compelling. There are many other reasons (p. 275) for a customer to love you despite your failings (p. 1559).

These "killer delighters" don't require universal appeal. If 15% of customers identify a feature as a primary reason for purchasing or remaining with your product, that's significant. When 15% feel that strongly, many more likely appreciate it, even if less intensely.

I quantify impact with this definition: A high-impact feature either (1) is regularly used by at least 51% of customers, or (2) is cited by at least 15% of customers as among their top three reasons for choosing or retaining your product.

This sets a high bar, but innovative, risky features demand a high bar. If you're undertaking projects that might exceed timelines or have uncertain outcomes, the potential reward must justify that risk.

Invest in leverage

There are some aspects of the business or product where small, incremental changes yield large results. It sounds too good to be true, but there are mathematical or structural areas where it is almost always true.

These include:

- Top-of-funnel quality
- Retention
- Pricing & pricing terms
- Onboarding
- Strategy

Each of these (and more) are justified in detail in this companion article (p. 1125).

Not included in the list above, is creating delightful, differentiated features. Those are special outliers, and therefore won't be produced by common rules of thumb. Still, it's almost always wise to invest a portion of your time on one of these asymmetric bets.

Maximize optionality

If we don't know how the future will unfold, we can make choices that maximize the options we have when we get there. More than flexibility, more than avoiding lock-in, building systems that are almost always ready to handle anything that arises.

Some examples:

- Keeping costs low enables all kinds of pricing and packaging while still being profitable, allowing for testing today and resilience in future.
- Selecting well-established, actively-development cross-platform libraries and frameworks for building user interfaces, so you're able to handle any evolution in platforms and devices.
- Plug-in systems, so that both you and your community can build things that you cannot imagine today.
- API-first architecture so that you own front-end tools, and your own back-end systems, and customer integrations, survives evolution.
- Wrappers around vendor services, so that you can swap out vendors if one becomes unstable, or too expensive, or lags behind others.

Some kinds of optionality require additional work today. For example, vendor-wrappers don't add any value today. Those techniques are wise for mature companies where stability and predictability are more important than releasing a feature a month earlier, but might be the wrong choice for early-stage companies who must rely on their velocity to win against incumbents (p. 295).

Portfolio of bets

Portfolios reduce variability at the expense of reducing maximum upside. That is, you're unlikely to have zero wins (so your downside isn't too bad), but wins have to make up for the losses, so even the occasional massive win isn't as massive as it would have been. The old joke is that the best investment portfolio would have been to buy Amazon at IPO and hold forever. Sure, but if you applied that advice to some other IPOs that year, you'd have $0. A portfolio of stocks means you'll never go to zero, but your maximum growth will be far less than best stock in the portfolio.

> **Mathematical sidebar**
> Why do portfolios work regardless of the underlying probability distributions? The Central Limit Theorem[1075] makes this precise: When you draw repeated samples from any distribution, then plot each sample's mean, the distribution of those sample means is Gaussian—a normal distribution—with a mean equal to the distribution's mean and a variance $\frac{1}{n}$ of the distribution's variance. So, total portfolio results are normally-distributed regardless of the underlying probability distribution, and we expect results near that mean, i.e. not zero, but also not near the maximum value.
>
> Even further, the The Lindeberg–Lévy Central Limit Theorem[1076] shows that the same is true even when each sample is drawn from a *different* underlying probability distribution. This holds only under certain conditions (independence, finite variance, and no single variable dominates all others). Arguably these conditions fail with distributions common in startup environments, e.g. some Power Laws have infinite variance.

Portfolios work when you want solid, predictable, but they don't work when you want outlier results. An example of the latter are venture capitalist or angels investor portfolios, where 65% lose money,[1077] and only 10%[1078] generate returns high enough to justify the risk and illiquidity. When hunting outliers, you need all-in investments (p. 867), not portfolios.*

Therefore, if the goal of your prioritization exercise is to find features that will be strong differentiators in the market and strong growth drivers, a portfolio is the wrong tool. On the other hand, if you're prioritizing a bunch of smaller things, where you want incremental but reliable results, a portfolio will get you those results. No need to argue about confidence.

Stop pretending you can quantify confidence, or even define it.

Instead, use techniques that work when the future is unpredictable.

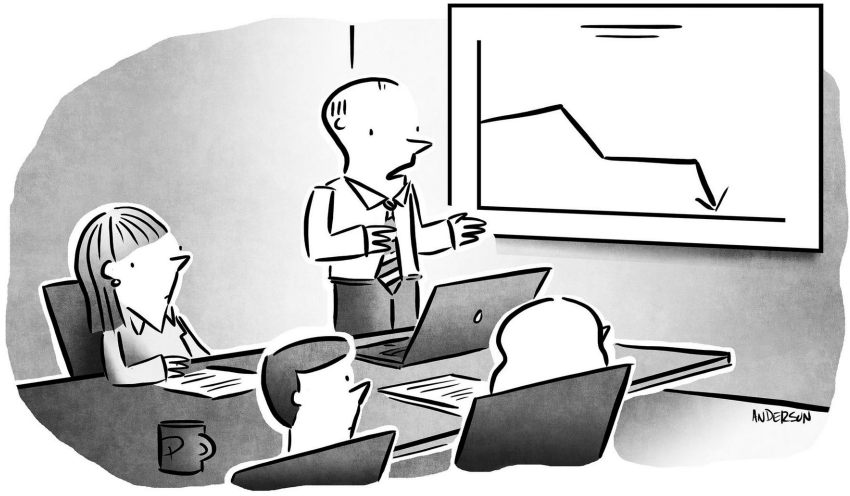
Because it is.

---

* Mathematically, the reason for this breakage is that the underlying distribution of startup returns is a Power Law that violates the Lindeberg criteria mentioned above.

Chapter 97:
Max MRR: Your growth ceiling

BUFFER · MAX MRR · MAX MRR AT BUFFER
NRR ≥ 100% · QUICK RATIO



"If by profits you mean an excess of revenues over outlays and expenses in a business enterprise over a given period of time, then yes, it doesn't look good."

credit 1079

For years I've battled the same, tired misconceptions:

- "7% cancellation is fine, especially for consumer businesses."
- "As long as I keep adding $300/mo of new MRR every month, I'll build a real business."
- "I'll keep scaling past a few million in ARR by doing what I've always done (p. 521)."

I've given real-world data (p. 335) to disprove these notions, but it's still hard to internalize and to apply it to your own companies.

Lately I've been using a new growth-related metric that I've called "Max MRR." It's easy to apply it to yourself, and because it's a visceral, tangible number—not some abstract financial ratio—it cuts through the veil of willful ignorance.

Here's how it works, along with a real-world example.

BUFFER

Buffer[1080] is a popular social media queuing tool, and a beloved company. Because their data is public[1081] (another example of their special, enviable culture), and they've gone through a few different phases of life, they are a perfect case-study of SaaS metrics. As we'll see, our subject metric of "Max MRR" accurately characterizes their business better than other commonly-used metrics.

Buffer started growing slowly, then quickly for years. Then revenue topped out (before COVID) and then shrunk (during COVID), reaching a new (lower) plateau in 2023. Then, after rebooting product, pricing, and target market in 2023, revenue started going up again (Figure 1).

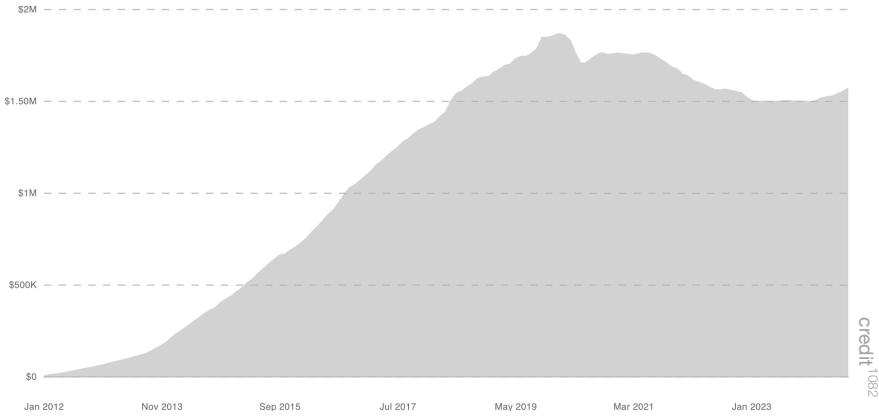


Figure 1: Buffer's MRR over time, exhibiting the Elephant Curve (p. 115).

What was going on in the customer base during these years when revenue peaked, shrank, and stayed constant? Was the customer base itself static? Not at all (Figure 2). During Buffer's period of stagnation and shrinkage, it was experiencing more new customers and upgrades than ever... but also even more cancellations and downgrades than ever. Extremely dynamic, but flat nevertheless.

The "Max MRR" metric is able to predict revenue, months ahead of time. We'll first use a toy example to see how it works, and then apply it to Buffer.

THE MAX MRR METRIC

Let's explore a simple, hypothetical SaaS company, with stable growth metrics, as follows:

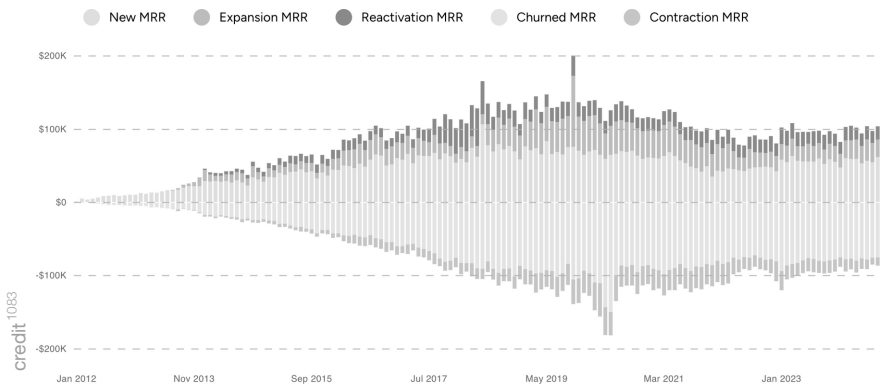


Figure 2: Buffer's period of stagnation and shrinkage contained "more of everything" — new, cancels, upgrades, and downgrades.

- Starting at $0 MRR
- +$1,000 new MRR each month from marketing and sales
- 5% cancellation each month from existing customers

Tracking MRR over time, it starts out growing nicely, reaching $10k MRR in 15 months, but then growth levels out, and it never reaches $20k MRR, even after 60 months (Figure 3).

The reason this happens is that new MRR is chugging along at a constant pace (+$1000/mo), whereas cancellation is non-linear; it is proportional to how large the company is. The larger the company, the more *absolute dollars* of churn there is, even though the *rate* of churn is steady at 5%. As churn dollars grow, growth slows.

Eventually cancellation-dollars are just as large as new-dollars, and the company stops growing, reaching a steady-state (p. 1087).

What is this maximum size? Mathematically it's trivial to compute; we just convert the previous sentence into numbers, where c is the monthly cancellation rate:

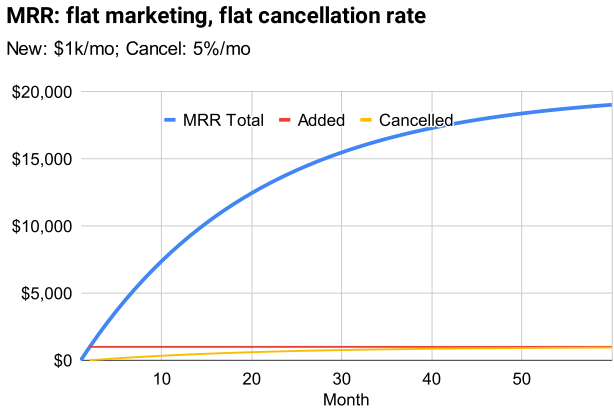


Figure 3

$$\text{MRR}_{\text{total}} \times c = \text{MRR}_{\text{new}}$$
$$\text{MRR}_{\text{total}} = \frac{\text{MRR}_{\text{new}}}{c}$$

In our toy example, $\text{MRR}_{\text{new}} = \$1000$ and $c = 0.05$, so the maximum total revenue is $\frac{\$1000}{0.05} = \$20{,}000$, which is exactly what the revenue is approaching in our chart (Figure 4).

This model also explains the common phenomenon seen in new startups: Fast initial growth, that tapers off sooner than the founders expect, given that the target market is still many orders of magnitude larger than the company. Since "market size" is not the ceiling, what is causing the slow-down?

The chart shows the answer. When "current MRR" is far away from "Max MRR," MRR will grow quickly; without many existing customers, there's not much churn. As customers accumulate, we churn grows; it's the *churn* that's slowing growth, not the *market*. As MRR approaches Max MRR, this is the dominant effect in growth.

Thus the Max MRR metric indicates not only the maximum size of the company, but the size at which growth will start slowing. Obviously we want this metric to increase over time, as we increase new

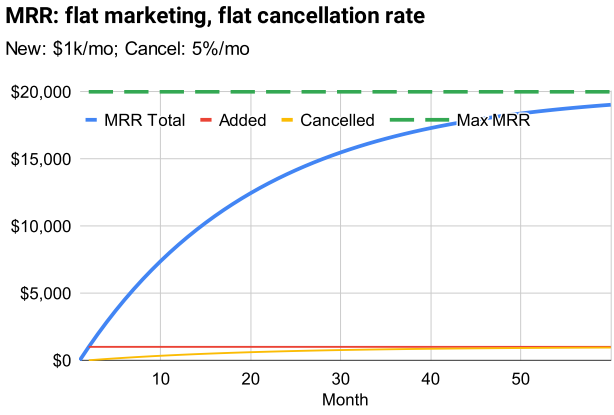


Figure 4

revenue or decrease cancellations. How can we know where we should invest our time?

Max MRR as cancellation rate improves

In real companies, neither new MRR nor cancellation rate is constant, so Max MRR changes over time. How does that look?

Varying cancellation even by small amount yields a large difference, because cancellation is a nonlinear—exponential in fact—so small changes are magnified.*

Let's take the same company, still with a fixed $1000/mo in New MRR, starting with 7%/mo cancellation for the first 20 months, then improving to 4% over the next 10 months, then holding at 4% (Figure 5).

* This effect—small changes in input creating large changes in output—is called "leverage," and this is why cancellation is one of the most important levers of a business. There are others (p. 1125).

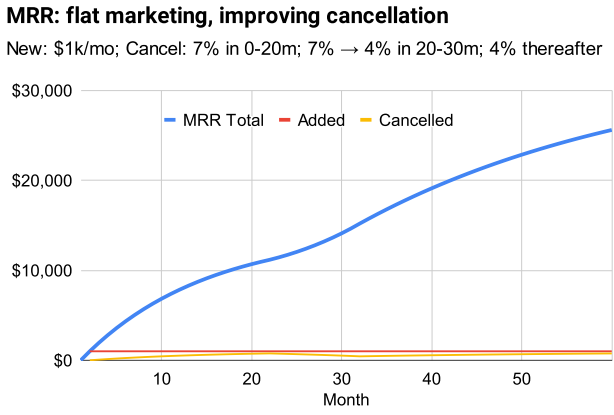


Figure 5

Notice that although cancellation changed relatively quickly, revenue doesn't react quickly, looking mostly linear but "wavy." It is therefore difficult to tell, looking at revenue, that we've made a massive improvement to the business model. In contrast, let's also plot the "Max MRR" metric (Figure 6).

Notice how the "max revenue" line shoots up as soon as cancellation rate starts improving, while MRR lags behind. When MRR gets near that ceiling, the cancellation rate dominates the (slowing) growth; when we lift the ceiling, "New MRR" becomes the dominant factor again.

"Max MRR" is therefore a *leading* indicator of long-term growth, because it shoots up quickly when conditions change, whereas "Current MRR" is a *lagging* indicator.

Max MRR as new MRR accelerates

Now let's swap which variable is improving; we'll fix cancellation rate back at 5%, and allow "New MRR" to increase.

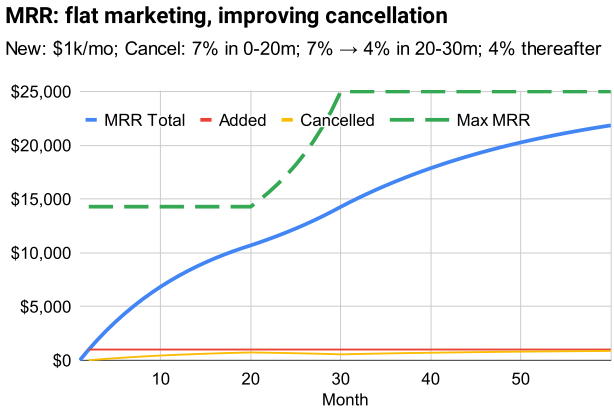


Figure 6

We already know that if we have a steady input from marketing, growth levels out quickly. What if marketing improves?

Suppose in the first month we add $100 in MRR, in the second month we add $200, and so on, adding $100 more each month than the month before. This is *accelerating* revenue—growing faster and faster. And we'll let it accelerate for 60 straight months (Figure 7).

The first thing we notice is that growth doesn't stop. "Max MRR" also grows alongside actual MRR.

But before we celebrate too much, realize that MRR is *not accelerating*. MRR is growing at more or less a constant pace—it looks like a line for most of the period in question. Yet this is a company that is *accelerating* their marketing output!

So this is disappointing. *New MRR* is accelerating, yet *MRR* is growing only linearly. This is once again because cancellations are exponential, which means cancellations are accelerating too. The acceleration from cancellation cancels out the acceleration from New MRR. Thus, even companies with accelerating new growth still typically grow revenue linearly, as analyzed here with many real-world examples (p. 115).

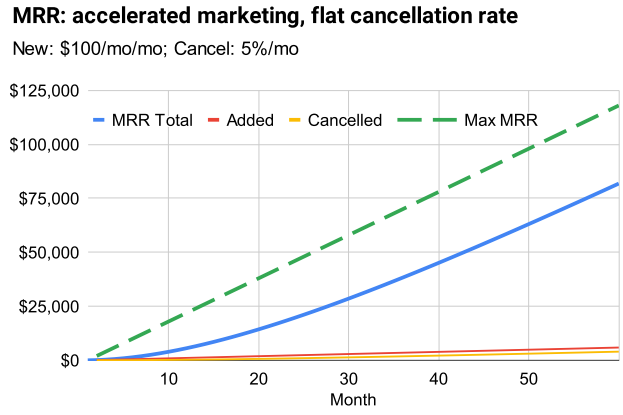


Figure 7

Furthermore, this scenario is still too optimistic; you cannot grow a marketing channel forever. In fact, channels not only top out, but tend to decline once mature (for reasons again detailed here (p. 115)). In a more typical situation, the company figures out a marketing channel, but then reaches channel capacity. This results in initially-accelerating growth, followed by the ceiling that we've now come to expect. And once again, the Max MRR curve presages this, and therefore is a useful leading indicator (Figure 8).

We also see the emergence of the classic "S-curve" revenue line. Growth accelerates during the combination of "not many customers yet who could cancel" and "new marketing channel is still growing." When marketing effectiveness levels off, the curve switches into deceleration. This has been observed thousands of times across companies in all industries and decades.

The Max MRR curve predicts this, months or even years before it happens. This is its utility.

Now that we have a good sense of how everything works with toy models, we're ready to apply this in the real world.

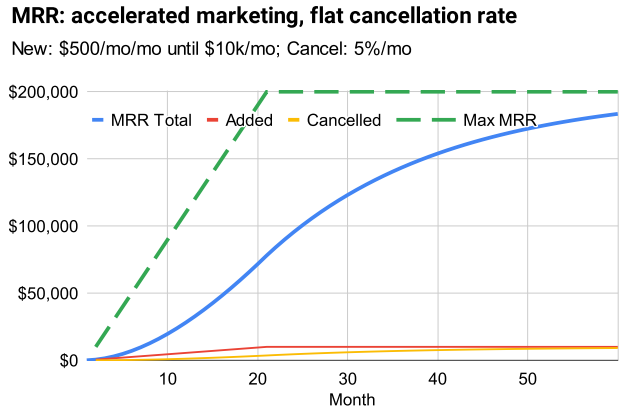


Figure 8

MAX MRR AT BUFFER

Back to Buffer. Figure 9 shows their MRR alongside Max MRR.

Buffer went through a few phases:

Timeframe	New MRR* & Cancels**	Discussion
2012-2014	MRR $3k → $40k Cancel 8% → 6%	As they found Product/Market Fit, they got cancellations under control, and accelerated new MRR, resulting in Max MRR increasing and revenue starting to accelerate.

* Includes new customer MRR, upgrade MRR, and reactivating previously-cancelled customers.

** Includes customer cancellations and downgrades.

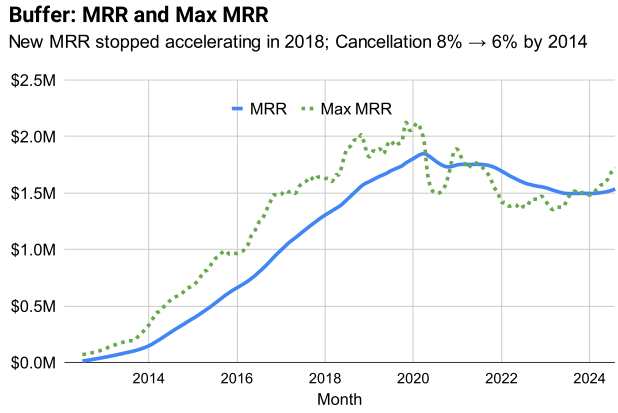


Figure 9

Timeframe	New MRR* & Cancels**	Discussion
2014-2020	MRR $40k → $140k Cancel 6%	Although cancellations never improved, they accelerated new MRR, with a few clear "found a new channel; channel saturated" events visible where Max MRR briefly plateaued. This maintained linear growth, but never accelerating growth, because cancellations were still high.
2020-2024	MRR $140k → $90k Cancel 7%	Revenue sharply declines, and struggles for years; cancellation pops up another percentage. New MRR is still large, but because cancellation is also large, Max MRR dips below revenue, so revenue falls.

* Includes new customer MRR, upgrade MRR, and reactivating previously-cancelled customers.

** Includes customer cancellations and downgrades.

Timeframe	New MRR* & Cancels**	Discussion
2024-	MRR $100k Cancel 6%	New MRR barely increased, but cancellation improved, so Max MRR has gone back above MRR, and we see a little growth.

Notice how the Max MRR line reacts much quicker, and with much larger magnitude, than any other metric. Anticipating both the good news and bad news.

Looking at the components of Max MRR, the most impactful culprit is the high cancellation rate. 6% is precarious, requiring forever-accelerating new MRR, and quickly tanking revenue as soon as new MRR didn't keep up, which of course it inevitably cannot.

What if Buffer had focused on customer retention, shifting the cancellation rate over five years from 8% in 2014 to 3% by 2019? Then revenue would have doubled (Figure 10).

Of course this is easy to predict knowing how Max MRR works. If you halve the cancellation rate, you double Max MRR. In 2023, Buffer founder and CEO Joel Gascoigne [1084] wrote a strategy letter [1085] saying the same thing. He shifted the corporate strategy back to their ideal customer (p. 317) inside their best target market (p. 1211) (entrepreneurs and small businesses), back to a pricing model that worked better for them (freemium (p. 1385)), who therefore retain at higher rates. And it worked—we can see cancellation diminish, MRR increase, and Max MRR increase ahead of MRR.

* Includes new customer MRR, upgrade MRR, and reactivating previously-cancelled customers.

** Includes customer cancellations and downgrades.

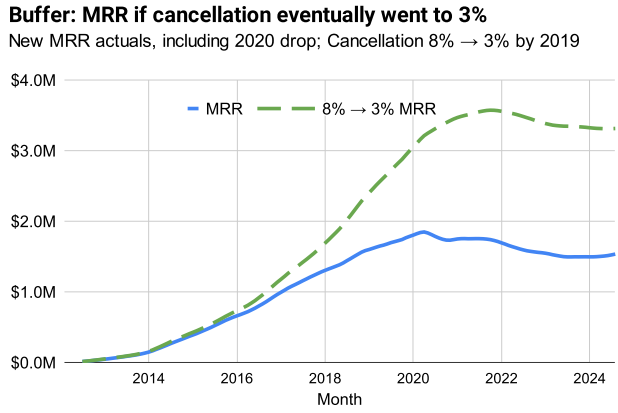


Figure 10

MAX MRR VS NRR

NRR is Net Revenue Retention, which answers the question: If no new customers were added, how would revenue change this month, as a percentage of current MRR? Specifically:

$$\text{NRR} = \frac{\text{Upgrade} - \text{Downgrade} - \text{Cancel}}{\text{MRR}}$$

When upgrades exceed cancellations and downgrades, revenue grows every month, even when you're not adding new customers. At what point would revenue top out in that scenario?

If the market were infinite, you would never stop growing! Therefore "Max MRR" ceases to have meaning.

Of course, markets are not infinite, just as marketing channels are not infinite, so you will top out anyway, but not because of the factors that go into Max MRR. Rather, it will be because you'll have reached saturation in your target market.

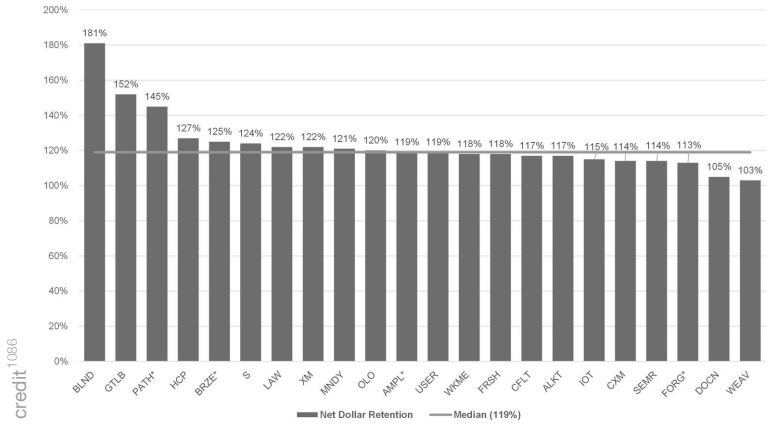


Figure 11: Average NRR at IPO for SaaS companies is 119%; all were above 100%. Average ARR at IPO is $255M. (Year 2022)

This does demonstrates a simple point: After high retention, the most powerful growth lever is to have NRR ≥ 100%. Indeed, almost all public SaaS companies have NRR ≥ 100%, exactly because if they didn't, their Max NRR ceiling would be too low for their lofty revenue goals (Figure 11).

This rule isn't important only for heavily-funded VC-backed companies gunning for hundreds of millions of ARR. It's just as valid for the solopreneur, because NRR creates growth even with a limited budget for marketing. It also means your customers are growing with you, becoming more successful, receiving more value, and happy to share some of that with you (p. 275) in the form of price. This is a wonderful signal and a wonderful business model regardless of your long-term financial goals.

MAX MRR VS QUICK RATIO

The SaaS Quick Ratio* is sometimes used in a similar way to Max MRR, measuring how much cancellations and downgrades are dragging down growth from new customers and upgrades:

$$\text{Quick Ratio} = \frac{\text{New} + \text{Upgrade}}{\text{Cancel} + \text{Downgrade}}$$

A high-growth, early-scale company should have a Quick Ratio of 3-4; a company at scale with healthy growth should be 1.5-2. Once it reaches 1, the company isn't growing, and below 1, it is contracting. This is similar to Max MRR.

However, useful as it is, it does not forecast revenue growth at all, as is evident from Buffer's Quick Ratio (Figure 12).

Buffer's Quick Ratio drops precipitously during Buffer's multi-year period of healthy growth. It continues its trajectory even when growth stalls and reverses. It starts recovering before the company actually made changes in 2023.

This doesn't invalidate Quick Ratio as a useful metric, it just isn't a predictor of revenue growth. Max MRR is better.

Also, Max MRR is easier to understand and get excited about. It feels tangible to have a "revenue ceiling" that you are trying to increase. It's hard to get excited about a goal like: "Let's get our Quick Ratio from 1.4 to 1.6."

No metric tells the entire story on its own; Max MRR is no exception.

---

* Easily confused with the financial Quick Ratio,[1087] which is the ratio of ready-assets to liabilities, used to measure how easily a company will be able to cover its financial obligations.

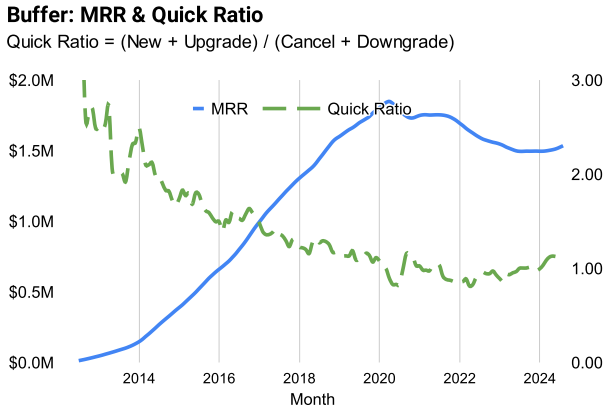
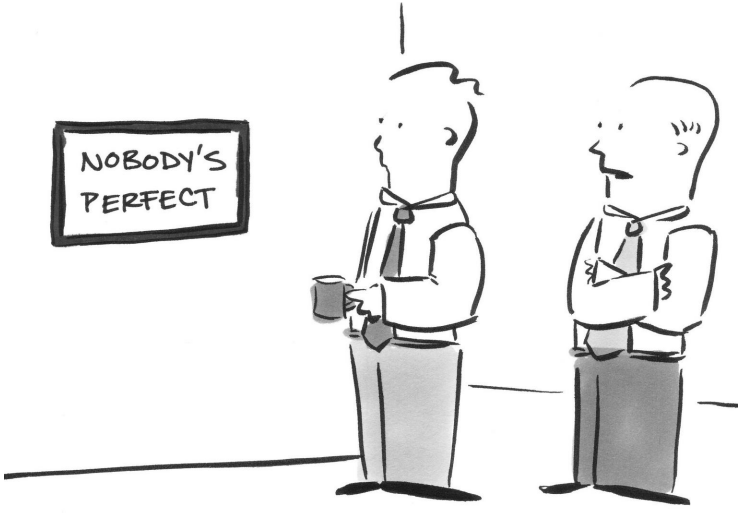


Figure 12: Buffer's Quick Ratio is uncorrelated with revenue growth.

But, it's a fun, simple, explanatory, predictive, and concrete number that is applicable for companies of all stages and sizes.

Chapter 98:
Mission, Vision, poTAYto, poTAHto

WHO CARES? · PURPOSE, VISION, MILESTONE
PURPOSE-DERIVED



"You're right. It's not much of a mission statement."

WHO CARES?

Mission, vision, purpose, BHAG, North Star. Are they useful, or academic nonsense?

For many of us, these are highfalutin terms that have no role in an early stage startup, because we're too busy making stuff and realizing that customers wanted something else. By the time the company is large, there are teams of people—probably in marketing—carefully sculpting these sentence fragments in large serif fonts on "About Us" pages that no one reads and no one believes.[1089] And by no one, I mean not employees, not customers, and not investors. Phrases that sound grand but are just grandiose.

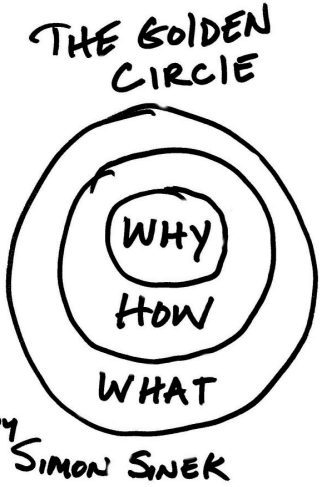
How can these words matter, when pundits can't even agree on their definition? Take "mission." One interpretation is like "missionary"—our higher purpose, something bigger than ourselves, that we are helping to bring about. So Patagonia's mission[1090] is "to save our home plant," though what it *does* is sell outdoor clothing. Or Tesla's mission[1091] is "to accelerate the world's transition to sustainable energy," though what it *does* is sell cars, followed by selling batteries and solar panels. Or Coca-Cola's mission[1092] is "to refresh the world in mind, body, and spirit," but what it *does* is sell barely-potable chemicals and containers of said chemicals embedded in carbonated water. Well, two of those three companies are at least fulfilling their mission.

The other definition of "mission" is nearly the opposite: Your current execution goal. An army battalion has a mission to conquer and defend a region. A starship has a five-year mission to explore new worlds while not interfering *(wink wink)* with new life and new civilizations. McDonald's mission[1093] is "to be our customers' favorite place and way to eat and drink," which is about themselves and how customers regard them rather than something bigger than themselves. FedEx's mission[1094] is "To produce superior financial returns for its shareowners [sic] by providing high value-added logistics, transportation and related business services," which is unabashedly about stock price and summarizing the service they provide, not a higher purpose. Points for honesty, but not inspiring.

So, it's unclear what "mission" or "vision" ought to mean, and unclear whether they have utility. And yet, some of the greatest and most well-respected and most admired companies in the world are serious about their words and have been from the start. Khan Academy[1095]—"Our mission is to provide a free, world-class education for anyone, anywhere"—and they have. TOMS shoes[1096]—"We're in business to improve lives"—and they have done so for many millions of people.[1097] DuckDuckGo[1098]—"Showing the world that protecting privacy is simple"—prioritizing not just the privacy of their own users but advocating for privacy rights online. And the aforementioned Patagonia, where the founder placed the company into a trust to ensure that their history of protecting the planet would forever also be their future.

Is it smart to adopt this "bigger than ourselves" definition of "mission," and to genuinely have one? This question has been explored at length, perhaps most famously by Simon Sinek in Start with Why

(2011).[1099]* He argues** that having a higher purpose makes companies better on every front:



credit[1101]

- **Loyal, vocal customers** spreading the word, even if prices are higher and some desired feature isn't implemented, because the company is meaningful to them, even part of their identity.***
- **Passionate employees** with less turnover, because they're working for a purpose, not only a paycheck.
- **Differentiation** in a market full of only-profit-driven companies that exploit employees and customers, which consumers hate (today even more so and more vocally than ever).
- **Resilience** during economic downturns and negative publicity as purpose transcends price and occasional missteps.
- **Consistency** in messaging, prioritization, and goals, which is useful for every company, but creates a special clarity and justification.
- **Raising money** from angel investors, who realize you'll probably fail, but who are at a phase of their own lives where they wish to make a difference.
- **Higher profits****** as a result of the factors above, which yields (perhaps ironically) a greater return to shareholders.

---

* Although with 67 million views, more people have received his ideas through his TED talk.[1100]

** The list below is his talking points together with those made by other researchers.

*** See "Willingness-to-pay" (p. 275) for more.

**** There is little empirical evidence to support this common claim. There are plenty of anecdotes, including some from this article, but almost none of the largest 1000 companies in the world are mission-driven, and a study showed[1102] that

- **Legacy** (p. 561) that outlasts any one person or product, providing meaning and fulfillment in a world where otherwise "work is just work."

You don't have to be a mission-driven company to succeed. In fact, most successful companies aren't. They tacitly adopt the executional definition of "mission" and ignore their mission anyway. You can do that too. Maybe you should; you just want to make a dollar and get out of here.

Even if so, I think you can be mission-derived, rather than mission-driven. Just because it's wasn't your origin story, doesn't mean you don't care about having a positive impact on the world, which you can declare and intentionally advance. More on that later.

PURPOSE, N-YEAR VISION, NEXT MILESTONE

Because "mission" and "vision" and other such words are untethered to consistent definitions, I prefer words that more closely connote what we intend.

The following is what I like.

---

while "purpose + execution" resulted in superior returns, "purpose alone" did not. There is weak evidence[1103] that "purpose-washing" (claiming to be purpose-driven, when you are not) results in lower performance, however this correlation might suffer from hindsight bias.

Purpose

Our Purpose is the transformation in the world that is bigger than ourselves, that we hope everyone involved is excited to help bring about.

This is like Khan's "Everyone deserves a free, world-class education" or Tesla's "Accelerate the world's transition to sustainable energy."

We will never fully achieve it, therefore it is not a "goal in our operational plan," but rather it is answers: "Why are we doing any of this? Why should anyone else care?"

The Purpose is not about us, but about everyone else. It doesn't mention us or what we do. It is important to us and to others, even this company didn't exist. Transitioning to sustainable energy is a Purpose no matter what companies come and go.

If others adopt the same purpose, we are happy. That isn't competition, that's an ally.* Exactly because the Purpose is bigger and more important than ourselves, we want others to join our Purpose.

N-year Vision

Our N-Year Vision is our future state: What we will have achieved in the market.

Here "N" should be smaller for new companies, larger for established ones. The reason: It must detail a situation that is different from today, so that it articulates an evolution, but not so far in the future that it isn't actionable today. Rule of thumb: Divide the age of the company by 3, and round up.

It must include:

---

* Examples: Khan Academy gives away its AI teaching assistant to help other educators build curriculum. TOMS shoes shares its R&D on sustainable materials with other shoe manufacturers. Tesla open-sourced the patents[1104] to its core battery technology, even though "better batteries" is one of their fundamental inventions. Netlify coined the term "jamstack"[1105] to promote a new server architecture that rivals not only adopted, but eclipsed Netlify in revenue.

What did we build?
The product or service. Tesla started with[1106] the first battery-powered luxury sports car. Patagonia started with[1107] hand-build equipment for climbers; its initial clothing line was just T-shirts. Amazon started with a vision of "Earth's Biggest Bookstore."

For whom?
What is the subset of customers that we will be serving? Can be larger than the ICP, but cannot be "everyone." Tesla started with rich Americans, Patagonia with serious American rock-climbers, Amazon with only 2,000 titles.

The insight
What is the special thing that differentiates the company, either in the resulting product or in how it will achieve it? This is the critical thing that makes this company "this company." Tesla's insight was that a new kind of battery could result in performance that matched the fastest gas-guzzling cars. Patagonia's insight was that climbers are also nature-lovers, and would appreciate not only their commitment to that cause, but products with minimal impact on the environment, and durable so that you minimize landfill contributions. Amazon's insight was operational: to eliminate both physical stores and hold almost no inventory, allowing them to scale faster and have lower prices than physical bookstores. Famously: "Your margin is my opportunity."

Because it must be brief (one or two sentences) it must necessarily be reductive—so over-simplified that it lacks precision. Therefore it is the starting point for a fuller explanation. The strategy document is where you specify those details. The short summary aids in comprehension and communication.

And so, in my own words:

- Telsa: Create the first high-performance luxury sports car for Americans that is powered only by batteries.
- Patagonia: Build the most durable, low-environmental-impact equipment for serious rock-climbers.
- Amazon: Create the largest and most affordable bookstore on Earth, by selling exclusively online with just-in-time inventory.

credit[1108]

"I'm a doer! I get results! I make things happen!
I'm intentionally vague!"

The Vision should be clear, concise, declarative, and describe a specific future. Be bold, but plausible.

Next Milestone

The most important thing to achieve next, and how we'll know when we've arrived.

Generally you can identify the next milestone by working backwards from the N-Year Vision. It looks something like:

> If we're going to achieve [final result] by [far-future-date], we will have to accomplish [1-5 critical projects or results]. So, right now what we need accomplish next is [near-term project], and we'll know we've achieved it when [objective observable].

Some examples, with companies at different stages and founders with different goals:

- I want to raise $1M in 10 months, so I can hire my previous team who is ready to join up. So, I will need to begin raising in 6 months, and to convince investors (and myself!), I will need to have proven that people want to buy our software, and that the market is large and growing. If I can get 200 paying customers by then, I'll have proved that people want it. But today I have only beta-testers (although they are happy!). Therefore, the next milestone is to get 20 paying customers—enough that it's not a fluke, but going from even zero to one will take real work.
- I've reached $5M in ARR but growth has stalled. I've analyzed why it stalled (p. 1191): cancellations are too high. Because customers cancel at 5%/mo and my marketing is already good, I mathematically cannot grow (p. 1153) without reducing cancellation. I'm discovered that some customer cohorts have only 2% cancellation, and also pay 5x as much! Therefore, I need to evolve the company to target those types of customers with both marketing and product. I'll keep a lower-priced version, but the mixture of customers will change. Today the mixture is 10/90 in the wrong direction. We can't change that over night, so our next milestone is to shift *new customers* to a 35/65 mixture. That will represent significant progress, and after that we'll be a lot smarter about what to do next.
- It's been 5 years and I'm burned out (p. 399). I don't want to leave, because this is my life (p. 1005), and I don't want to do anything else. But I can't go on like this. The company needs the leader to be healthy, and also needs someone who has lead a company from my current stage to the next one; I'm not that person, neither in skills nor in emotional desire. In two years I want to have tripled revenue and tripled

profit, but I want someone else to help get there. Therefore, my next milestone is to hire a new CEO and manage the transition process with her* and the rest of the company, as well as my own, so I can stay for another ten years without burning out.

- Our business has matured into a sustainable level of growth, but despite many efforts, we can't grow revenue faster than it is. Our unit economics are great—CAC is low (p. 1377), cancellations are low, NPS is high, and so on. So, we need to expand into an adjacent space (p. 793) to accelerate growth. We don't yet know how to expand, though, so we're going to brainstorm some ideas from the article just referenced, then run experiments to figure out the best course of action, before committing millions of dollars and thousands of person-hours to the effort. The next milestone is to have run five experiments and selected the one or two things we're going to do.
- We've bootstrapped our indie game studio for three years and released two moderately successful titles. We're profitable but it's still hand-to-mouth, and we're tiring of the constant pressure to ship. Our vision is to have enough runway to take 18 months to build our dream game—something ambitious that could break out and define us. To get there, we need $600K in the bank. Our current games generate about $12K/month in profit, so we need to either boost that to $35K/month or find another revenue stream. Therefore, our next milestone is to create a small, high-margin DLC for our most popular game that can ship in 4 months and boost monthly profits to at least $20K—enough progress to make the bigger goal feel achievable.
- Our agency has grown to 35 people over the past decade, with solid enterprise clients and consistent 15% year-over-year growth. But I've watched SaaS founders build massive value while we're

* For those of you irked by my use of the feminine pronoun, I was personally in this position (p. 399) at WP Engine, and her name is Heather Brunner. I call her our "late-joining co-founder," because there's no better description for what has unfolded over the past ten years.

still trading hours for dollars. I want to transform part of our agency into a productized service with recurring revenue that could eventually become a software product. In three years, I want recurring revenue to be 40% of our business. But I also fully realize why this is so hard to do (p. 701). My next milestone is to identify our three most repeatable, high-margin service offerings, package one as a fixed-price monthly subscription, and sign our first 10 recurring customers even if it means giving them a steep discount to pioneer this model with us.

- Our open-source project has gained significant traction with developers (50,000 GitHub stars), but we're still working on it nights and weekends while holding down day jobs. We want to make this our full-time focus by creating a sustainable business around the project without alienating the community. In two years, we want to have a team of 5 full-time maintainers funded entirely by the project. The next milestone is to launch a hosted version with premium features that don't exist in the open source version, sign up 50 paying customers, and reach $10K MRR—enough to justify two of us quitting our jobs to work on it full-time.

- I acquired a 15-year-old traditional manufacturing business with a reliable customer base but antiquated operations and virtually no online presence. My vision is to double revenue in three years by modernizing operations and expanding sales channels. The first order of business is cost reduction through operational efficiency—we're leaving 20% of potential profit on the table through waste and outdated processes. Therefore, my next milestone is implementing a modern ERP system, retraining staff, and reducing our COGS by 12% within six months, which will generate the cash needed to fund our expansion into e-commerce.
- We're a Series B AI startup (p. 419) that grew rapidly by providing custom solutions to Fortune 500 companies. We're at $12M ARR with 90% gross margins, but building custom solutions isn't scalable and our growth is plateauing. Our investors are pushing for a more standardized product approach that could reach $100M ARR. Our next milestone is to extract the common patterns from our custom work, build a self-serve platform that addresses 80% of use cases without customization, and get 3 existing customers to successfully migrate to it—proving both that our platform approach works and that we can make the business model transition without losing our current base.

Note that in all cases, there was something specific to do, and some objective way of knowing whether it was done. Often the "objective way" is a metric (p. 645), but not all important things are numbers. Unlike the Purpose and Vision, the Milestone is where you get specific, tactical, operational, decisive, and either metrics-driven, or at least metrics-influenced.

To fully operationalize the Milestone, use this practical quarterly strategic planning method (p. 1065), then use this version of Rocks/Pebbles/Sand (p. 221) to prioritize work, and track progress using a sensible set of metrics (p. 645).

"I don't get it! We've got a mission statement, a credo, *and* a mantra!"

credit 1109

PURPOSE-DERIVED

You might be purpose-*driven*, or purpose-*derived*. Purpose-driven means the origin and true number-one goal of the company is to fulfill its Purpose, evidenced by behavior where you advance the Purpose even when it means lower sales, lower market share, lower profits, and activity that other companies would label "a distraction."

Purpose-derived means you have a Purpose, but it is not your singular motivation. It *is* good for others, it *is* bigger than ourselves, but it is a by-product of our success, rather than the driver of it. I believe most companies fall into this category (when they don't fall into the category of simply being a net-negative in the world).

Take Smart Bear. I did not start that company with a higher purpose. I had an idea for a tool that would "data-mine" the history of a software project. It wasn't a very good idea, but it lead to a very good idea, which was a tool to help software developers review each others' work.* But it did have a great derived Purpose: To increase software quality. The proof of this derived Purpose is in its evolution in the

more than 15 years since I sold (p. 45) and left the business: It accumulated nearly a dozen tools relating to software quality, many open-source, all beloved by their customers. Even at the time, I had written a free book all about how to do reviews well, none of which required buying our software. This is a good Purpose.

Or take WP Engine. I did not start that company with a higher purpose. My blog kept crashing when I got on the front page of link-sharing site Hackernews, so I reached out to other bloggers to find out how they handled it. They said: "I don't know, but if you find it, tell me, because I need that!"** But it did have a great derived Purpose: To enable everyone to run a superior website. After all, if you don't have a website, you're invisible. And websites have to be fast (or Google won't rank you high enough) and scalable (or you'll crash when you get that awesome PR hit) and secure (or the bad people will hack your site). But 99.9% of business owners (large and small) and individuals can not and ought not need to become technology experts! And some people are trying to have a voice in unsafe places, speaking truth to power. They need someone to trust, so they can thrive online, so they have the freedom to create. This isn't just a nice story I tell myself to believe what we do makes a difference—we've been told this, unprompted, more times than I can count. This is a good Purpose.

Apple started out selling computers to hobbyists. It was 21 years before they developed the "Think Different" campaign, establishing that they cared about (in their own words), "The crazy ones. The misfits. The rebels. The troublemakers. The round pegs in the square holes. The ones who see things differently. … And while some may see them as the crazy ones, we see genius." This was Purpose-derived, but it's what we remember, and it's referenced repeatedly by Simon Sinek himself in *Start with Why*. Apple did not, in fact, start with why. But it got there.

---

* Here's more of that story. (p. 193)

** Here's more of that story (p. 9), and here's the method I used (p. 239) to validate whether that initial signal was truly validation.

> *"True happiness is not attained through self-gratification, but through fidelity to a worthy purpose."*
>
> *–Helen Keller*

Slapping words you don't mean onto an "About Us" page does not create a Purpose.

What does create a Purpose, is figuring out the positive difference you can make in the world, then writing that succinctly, then actually incorporating it into your strategy and positioning and goals, so that you gain most of the listed benefits above, and so that on your deathbed you can honestly say: What I did, mattered.

Or best of all, you can be driven by a higher Purpose from the start, and build sustainable organization that brings about some of that change. It's one of the most wonderful things you can do with the precious few hours you have on this Earth. We live in an era where it's possible to do good as well as become rich.

I wish you luck in both.

Chapter 99: Navigating layoffs

BOO HOO · REQUIREMENTS
WHAT GOOD LOOKS LIKE

OH BOO HOO, THE RICH CEO HAD A BAD DAY

I hesitate to write this because it's hard to have sympathy for leaders who execute layoffs while keeping their own jobs.* Yet layoffs happen, and there are better and worse ways to handle them.

Companies don't do layoffs capriciously. Sometimes economic situations change unexpectedly; for example, layoffs due to COVID-19 weren't due to a failure of leadership. But often it's true that leaders made bad choices, wasted investment dollars, or failed to build

* While this is often true, justifying the stereotype, it's common for public company CEOs to be fired together with or just after layoffs. In the past few years, these included the CEOs of Peloton, CNN, IBM, Disney, Bed Bath & Beyond, AMC Networks, Stitch Fix, WeWork, Slack, Sears, JC Penny's, Boeing, and Uber. It is more common with public companies than private, and more common with professional CEOs than founder-CEOs.

a profitable company. Sometimes public companies follow changing Wall Street values—for example shifting to value profit more than growth—at the expense of employees.

Whether leadership was at fault or not, the task of executing layoffs is incredibly difficult, though you should not and will not find sympathy from your employees or the world at large. Why should they—the leader is responsible, yet the leader somehow gets to keep their job. Even if the leader is part of the layoff, they got paid more and likely have a nice severance too.

There's also a common perception that leaders are just moving numbers around on a spreadsheet, and don't care about the human consequences. I'm sure those evil bastards exist, but in my experience with startups in the past few decades, I've never seen that particular thing. I see leaders who throw up from the combination of guilt, shame, and anticipation, who say it's the worst day of their life, and who are permanently changed. Again, not asking for sympathy, but not every regrettable action is motivated by evil.

In any case, layoffs will happen, and there are better and worse ways to execute them. For the benefit of those laid off, those staying, the mental health of the leaders, and the health of the company, here are my suggestions.

REQUIREMENTS

However you execute the details, I believe you need to satisfy all of the following conditions. Often founders fixate on a few but not all, and that's why it goes especially poorly.

Accept that people will be unhappy
No matter how you do this, people will be upset—both those laid off and those who remain. Many won't believe your explanations, and en-

couragement or grand vision statements will not assure them. Most people aren't really listening after hearing the news. These suggestions aren't about preventing anger and confusion—assume those will happen. This is about reducing turmoil, not avoiding it entirely. It will be bad; we're aiming for "least bad." Sorry, that's the job.

General Patton on the decision, Mother Teresa on the exit
This phrase from our wonderful CEO Heather Brunner is applicable to many human interactions, and especially here. Be firm in your decision but generous in the exit. Whoever's fault this is, the people who are laid off don't "deserve it," so be as generous as possible.

This means long severance, extended benefits, and career services for resume building and interview coaching. At minimum, find points of leverage: small things that make a big difference. For example, in America, setting someone's last day as April 2nd instead of March 30th gives them an entire additional month of health benefits. Find every little way to leverage kindness.

Focus on the survivors
How you manage the people who remain is even more vital to your company's future than how you handle those leaving. Those leaving aren't really listening—they know they've been unfairly fired and don't care about your corporate strategy and its rosy future. But those staying need to start rebuilding trust.

First, they need to see how generously you're treating departing colleagues. Do this because it's right, but also because it shows remaining staff that you're humane. You're doing what you believe is necessary, but doing everything you can to reduce damages to innocent people.

Talk specifically about the future this enables—not just better finances, but what you will execute to keep the company safe, healthy, and never go through this again. Maybe you're shutting down a failed second product while keeping the solid core business. Maybe you're reorganizing around more efficient processes everywhere. Whatever it is, be clear why this change creates future-proof safety. This had all better be worth it.

Tell everyone simultaneously
As soon as anyone gets even a small whiff of layoffs, 95% of the company will find out. The rumor mill moves at the speed of Slack, text messages, and Discord groups you didn't know existed, often bouncing through alumni as quickly as current employees. You do not want your message preempted by crowd-sourced worst-case scenarios.

This is why, while it feels impersonal, you must announce to everybody simultaneously. And the first sentence is that with an extremely heavy heart, you are saddened to announce that we are having a lay-off. Do not bury the lede. Of course face-to-face conversations are necessary, but you have to break the news without staggered times. During the announcement, explain that you (or appropriate senior management in the case of 800+ employees) have one-on-ones with each departing person and are also willing to meet with anyone else to talk it through.

In larger companies with management layers, tell everyone they should feel comfortable contacting any leader they trust, even outside their department or chain of command. These conversations are appropriate and expected.

Visual and live
Zoom if everyone is remote because you have to do it simultaneously, but of course in person is better. Anything asynchronous or impersonal is the worst: email or even a prerecorded video. The medium matters; those media scream: Management is uncaring, inhuman, and couldn't even be bothered to tell me to my face.

Cut only once
The very worst thing is a follow-on mini-layoff, even a year later. Any remaining trust will evaporate completely. You justified the first cut, in that it would create a sustainable future; if more layoffs follow, you've proven that you're either a liar or incompetent. Both are excellent reasons for everyone else to flee.

WHAT GOOD LOOKS LIKE

This is a real-world example of layoffs done correctly.

I'm keeping the company and founders anonymous—partly for privacy, but also because I don't want search engines or AIs to permanently link this company to a layoff that happened over ten years ago. The company not only survived but maintained their culture and continues to thrive today.

Their approach was methodical and compassionate:

The company divided employees into two groups—those being laid off and those remaining. Each group was placed in a different conference room, with a co-founder leading each room. The news was delivered simultaneously to both groups.

The general message was identical, but the details were tuned for the audience. For those staying, they outlined the services being provided to departing colleagues but focused primarily on the company's future health, strategy, and execution. For those leaving, they minimized talk about "how great this is for the company" and instead focused on benefits, reassurance, and support.

While I'm not sure if they took questions during these sessions, my personal style is to take Q&A, fully realizing that it opens you up to nasty comments and difficult questions. This is valuable, as it gives you a chance to hear harsh truths, stand before everyone and take full responsibility, and show you aren't dodging accountability.

When someone is especially critical, not only of the facts but of you personally, taking the high road while accepting responsibility shows you fully understand the weight and consequences. The more extreme their criticism, the more moderate and reasonable you appear by simply acknowledging the truth and explaining the path forward.

If criticism becomes personal ("leadership doesn't care about us"), respond with honesty: "I don't expect sympathy—I'm the one who got us here. But last night I threw up because I was so nervous about doing this. I'm crushed and embarrassed. I feel guilty and shame that we are

in this position. Some of you won't believe me and I'm not asking you to sympathize with me, but this is the hardest thing I've ever done. It's made me question whether I should be doing this job at all. Now I feel an overwhelming responsibility to ensure we succeed, so that in retrospect we will all look back and see that this sad day was necessary for a safe, sustainable future. I don't expect you to believe me—I'm just telling you the truth."

They were Mother Theresa on the exit. They provided six months of severance, one-on-one consulting services for each person to create a better resume, a strong cover letter, and interview practice. They paid for placement services to proactively help folks find another job before severance ran out. They wrote proactive letters of recommendation, without departing employees having to ask for them, so they could demonstrate this was because of company circumstances, not performance issues.

The aftermath? People who stayed actually comforted the founders, saying they understood how difficult this was, recognized the necessity, and appreciated that the company genuinely cared for employees. Dozens expressed this sentiment, even some of the people who were let go. You shouldn't have this as an expectation or a metric of "success," but if the culture is wonderful, the founders are genuine, the execution is crisp and humane, it will go as well as it can go.

Were others secretly angry or immediately job-hunting? I'm sure so. But this approach made the worst day of the founders' lives as "least bad" as possible. By being honest, vulnerable, taking responsibility, and moving forward with those who still believed in the culture and vision, they navigated an impossible situation with as much grace as possible.

Now, plan carefully, do what you must, but you'd better be right this time.

Don't let them down again.

Chapter 100:
What to do when growth tapers off

CHURN · PRICING · INTERNAL GROWTH · CHANNEL
MARKET · PURPOSE

Every company hits a frustrating phase where growth tapers off, or levels off, or even declines.

It happens whether you're the most popular "viral" social media company in history (Figure 1)...

...or one of the most-respected bootstrapped, transparent, customer- and employee-first companies in history (Figure 2)...

...or a programming language used by millions of people (Figure 3)...

...or people switching to mobile devices (Figure 4)...

The founder protests! This shouldn't be happening because:

- The product is better than ever *(more features, fewer bugs)*
- The market is bigger than ever *(more potential customers, more money being spent)*

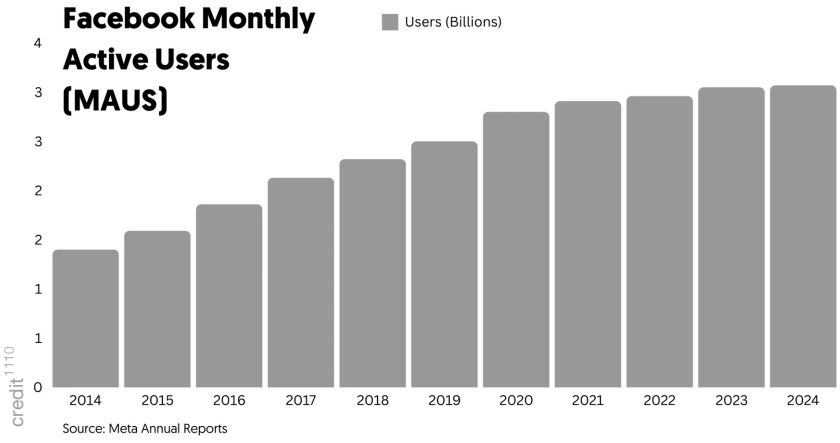


Figure 1

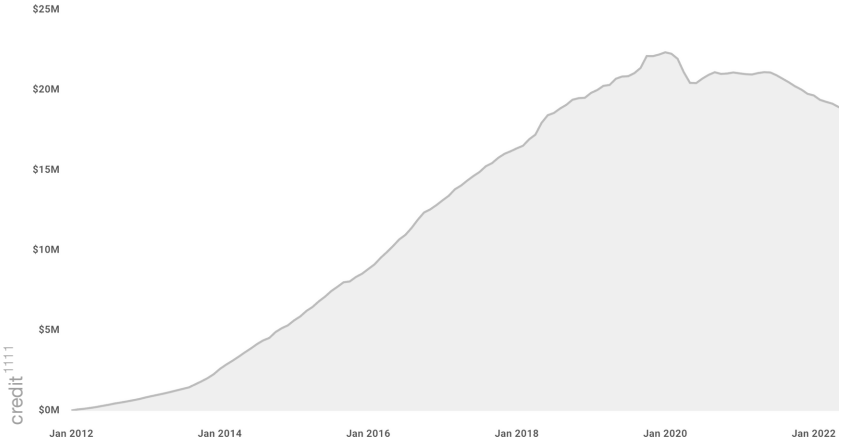


Figure 2: Buffer's[1112] ARR

- Our brand is stronger than ever *(more customers, more presence, more time)*

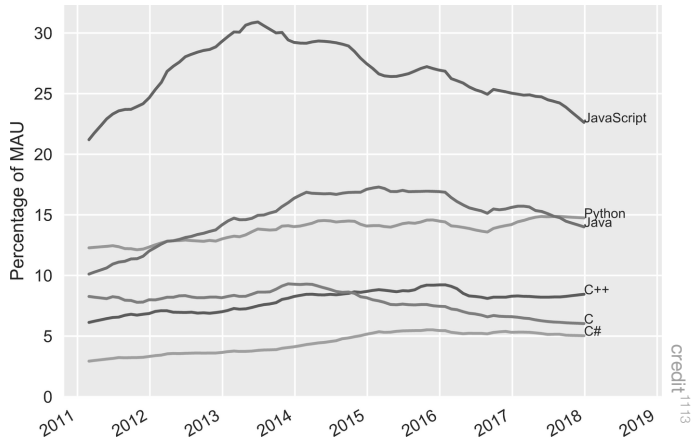


Figure 3: MAUs of major programming languages as a percentage of all developers

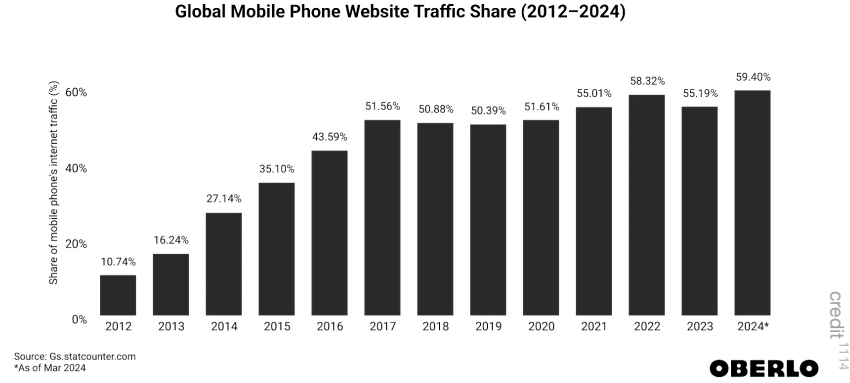


Figure 4

- We're smarter than ever *(more experience, more data)*

And yet, it's happening.

How do you fix it?

What definitely *won't* work, is doing (more of) the same things that got you here. Nor doing incremental things, because this is not an incremental challenge.

Rather, you need to diagnose the most significant force causing the asymptote, and attack that force with non-incremental effort.

> *"To beat the plateau we have to do things that we had previously considered out of bounds."*
>
> *—Laura Roeder*[1115]

I use the following framework to diagnose the underlying growth problem, whether it's for one product-line inside WP Engine, or for top-line revenue at another startup.

I ask the following questions, in this order, stopping at the first question that identifies a problem. These are ordered by causal power; that is, addressing a lower one will not solve the growth problem, if there's a problem higher up the list.

Q1. LOGO CHURN: ARE CUSTOMERS LEAVING?

"Logo churn" is the corporate way of saying "churn measured by number of customers." Not by MRR, not looking at upgrades, just the rate at which customers are leaving.

> **Sidebar: Calculating "churn rate"**
> A typical rate is "percent of customers lost per month," which at its simplest is $\frac{c}{s}$ where c is the number of customers who cancelled during that month and s is the number of customers at the start of the month. However this doesn't handle customers who signed up but then cancelled within the month. Therefore, it's better to either (a) include sign-ups using $\frac{c}{s+n}$ where n is the number of new customers added during the month, or (b) ignore those in/out customers, tracking that quantity separately (because you probably want to do something different about it!), and calculating c_s/s where c_s is number of customers *who were present at the start of the month* who cancelled within that month.

Logo churn is the worst problem because almost nothing else you do will make up for it. The customer is gone; there's no chance for recovery. This is often correlated with negative reviews and negative social proof, hurting future growth as well—a double negative.

The worst part is that customers are saying "I don't want this." That's a fundamental problem transcending "finance" and "business model," piercing the heart of what you're doing and why. If customers don't want the product, nothing else you do matters.

You made a promise that customers wanted—that's why they bought. But you didn't keep your promise—that's why they left.

Dangerously high logo churn rates are:

- B2C: 5%/mo

- B2B, Small Business: 3%/mo
- B2B, Mid-sized Business: 2%/mo
- B2B, Enterprise: 0.8%/mo

Small business owners love to push back on this, insisting that their 7%/mo churn rate is "normal for the industry" and "fine because we're a small company" and "hasn't been a problem so far."

Incorrect. And, because they refuse to understand why 7% churn is high, they're mystified at why it's so hard to grow.

The math is simple. To see exactly why this is wrong, see the "High Cancellation" section of this article (p. 335) for an analysis of the specific case of Buffer (above), and why high cancellation (5%, in their case) is the one and only cause of their revenue ceiling. Here's the summary (Figure 5).

The math is undeniable, as is the issue that customers aren't reliably getting enough value from the product. This is a fundamental product problem that must be solved.

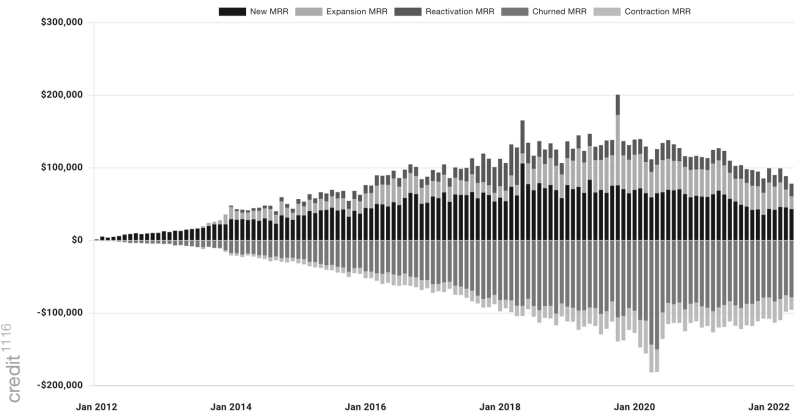


Figure 5: Buffer's unit economics; new-MRR reaches a natural maximum of new-customers-per-month, whereas cancellation never stops growing in absolute dollars, because it is proportional to the size of the customer base.

How?

You'd expect the answer to be: Talk to people who have churned, or are churning now. Unfortunately, they typically don't want to talk—they don't want to invest more time with you, and are often unhappy or otherwise emotional. Even when you do communicate—whether live or by survey—they often give a generic complaint (e.g. "not enough value" or "too expensive") instead of the real reason. Why not? Because it's easier, because they don't want to hurt your feelings, because they themselves haven't analyzed it carefully.

To see why these answers aren't useful, consider "too expensive". Remember, they signed up for the product, at this price. It wasn't "too expensive" then. Which means "too expensive" isn't the reason. It means there was some expectation that wasn't met; they *were* happy to pay that price for *something*. What expectation did they have, which failed to deliver? The product didn't work the way they thought? The product didn't do what it promised? They used the product incorrectly? They were never the right customer for it? The product didn't fit into a workflow? Their business changed? Your champion left the company?

At minimum, you need a thoughtful interrogation that gets to the root of the issue. Few (ex) customers will participate in such an activity; cherish those who do, and spend enough time with them honestly rooting out the issue.

Better: Look for signals that a customer is not being successful *before they cancel*, so you can reach out while they're still ready to talk. That means understanding what is correlated with cancellation, *and* what is *anti-correlated* with retention (because all customers have some behaviors in common).

Finally, remember that "retention" is something everyone in the company has a hand in, and thus needs to diagnose and address. If marketing attracts the wrong customer, churn will increase. If sales force-closes customers who aren't a fit, churn will increase. If support lets people down, even in the feeling of the interaction, churn will increase. If the product isn't intuitive or doesn't fulfill the promise,

churn will increase. Early in their journey, Hubspot addressed even more areas[1117] to halve their churn rate, which converted them from "barely holding on" to a "growth juggernaut."

This is difficult, but worth the effort.

Q2: IS YOUR PRICING CORRECT?

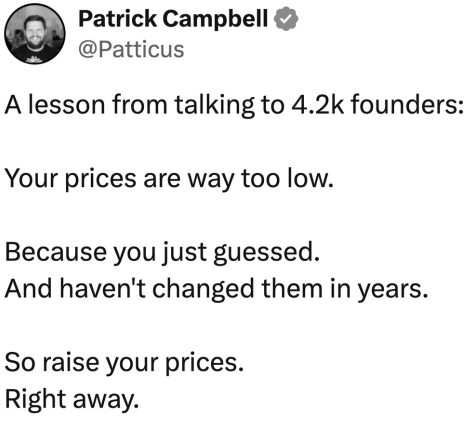
The answer is almost always: No.

Patrick Campbell explains why in Figure 6.

But also, the problem is almost always: It's really hard to know.

Even small changes in pricing can instantaneously increase revenue without changing how many people sign up every day, or how many people cancel every month. (Which means it dramatically increases profitability.)

Two examples from small companies (Figure 7) (Figure 8).



credit[1118]

Figure 6

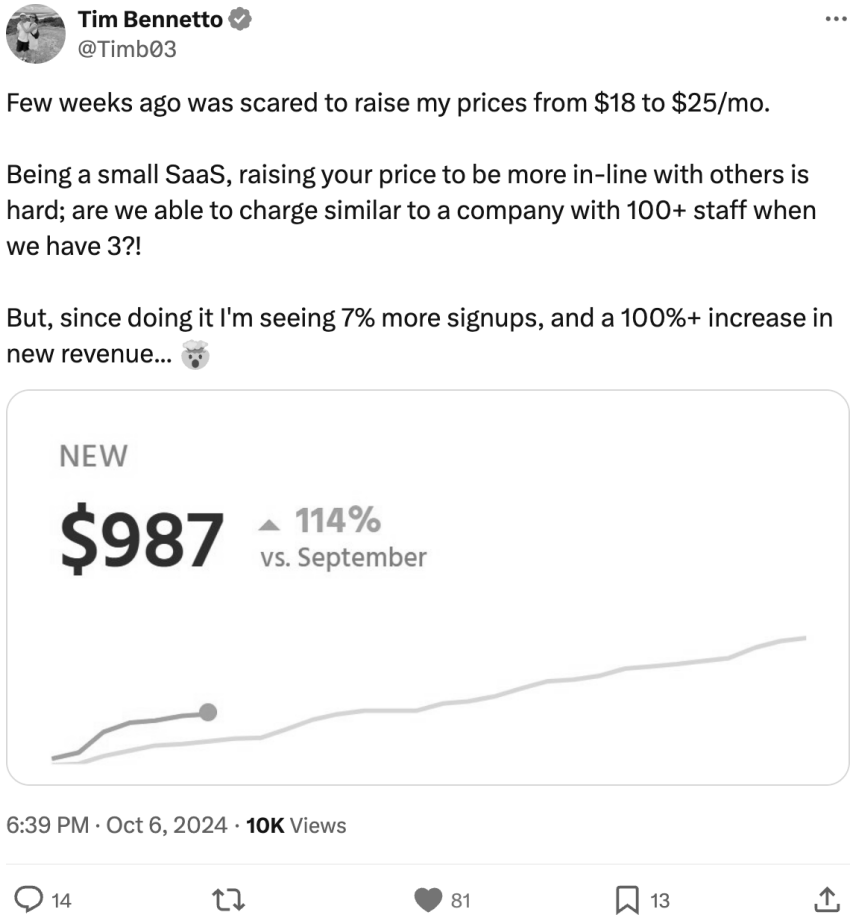


Figure 7

The obvious action is: Experiment. There are more systematic ways, but they're difficult to execute, and in my experience, even with expert assistance, half the time you get it wrong.

Indeed, price increases don't always work.[1121] When customers don't have more money, or demand isn't strong, or competition is strong, or you've built your reputation on having the lowest prices, it

credit[1120]

Figure 8

won't work. And if you increase too much, it's not the same business anymore (p. 515).

Yes, pricing is art as much as science. It's also one of the largest and most immediate levers you have.

Q3: ARE EXISTING CUSTOMERS GROWING?

The next step is to examine "net churn", which considers both cancellations and internal growth—how much revenue you're generating from existing customers through increased usage, or tier-upgrades, or additional add-ons.

Many small companies overlook the importance of internal growth, because when you're small, there's very little existing customer base to grow, so it's smarter to focus on getting more customers instead. But

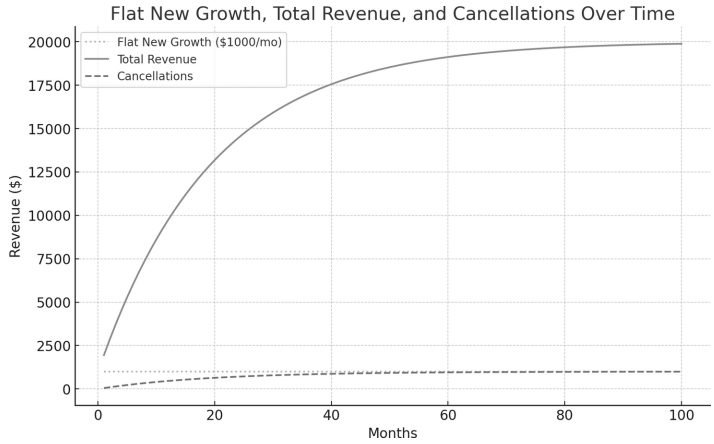


Figure 9: A model with $1000/mo of new revenue and 5%/mo cancellation. Growth tapers off as cancellations outstrip new customers.

that form of growth inevitably slows as the company grows, so your focus has to change as the company changes.

The mathematical problem is that new-customer-acquisition grows with marketing, which *is not* exponential (p. 115), but logo-cancellation is a percentage of your size, which *is* exponential. As a result, plotting "new-versus-cancel" generally looks like this (Figure 9).

While new-customer-acquisition will never scale with the size of your company, a different growth vector does scale that way: Upgrades from existing customers. If 5% of customers upgrade per month, that counteracts 5% cancelling per month.

The modern metric for this is NRR (Net Revenue Retention). It is the annual revenue percentage change due to cancellations and downgrades (negative) and upgrades (positive). So e.g. with 30%/year cancellations (equivalent to 3%/mo), 10%/year downgrades, and 50%/year upgrades, all measured in ARR, we compute:

```
NRR = 100% - 30% - 10% + 50% = 110%
```

Your goal is `NRR > 100%`. This means you're not just compensating for churn, but growing even if you never added another new customer.

For many at-scale companies, `NRR` growth is *greater* than new-customer growth (Fastly, for example). Dropbox saw significant growth by introducing tiered pricing and premium features that encouraged existing users to upgrade. This boosted their NRR and propelled their growth (Figure 10), even when new customer acquisition slowed (Figure 11).

Often the problem is that you don't have a pricing mechanism that encourages upgrades. The classic "best" way for a SaaS companies is to price along two dimensions:

1. **Usage** (e.g. number of users, amount of data, tickets/month)
2. **Functionality** (e.g. "tiers" that include more features, integrations, service, compliance)

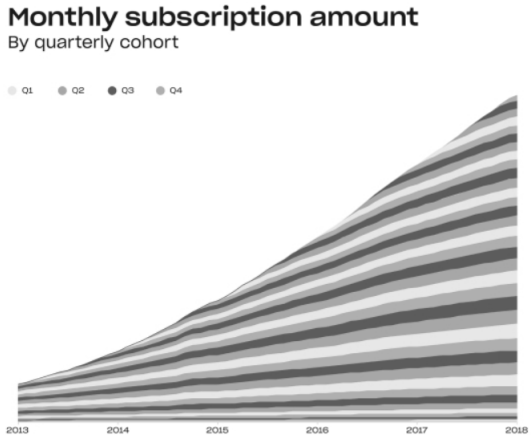


Figure 10: Dropbox customer cohorts grow faster from internal upgrades than they shrink from cancellation.

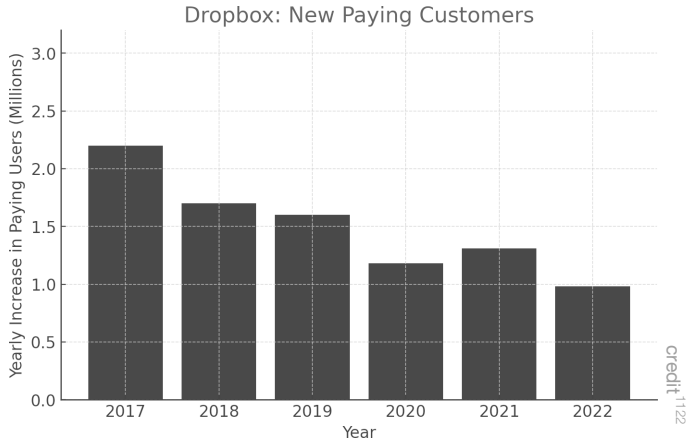


Figure 11: That's why Dropbox revenue continues to accelerate, even though the rate of adding new paying customers continues to decelerate.

For example it makes sense for Support Ticket Desk Software to charge more if more reps are using it (usage), or for additional features like AI responses (functionality).

The rule is that customers ought to pay more only when they're getting more value. Then it's fair and sensible, being at least Utility, not contrary to Love, and certainly not Coercion (from my WTP framework (p. 275)).

Often price-tiers segment the customer base, both by their budgets and their requirements. Large companies are willing to pay more for integrations and a higher tier of support. But customers don't often change segment (e.g. rarely does a small business grow so much that it becomes a large enterprise), and therefore this doesn't help you with upgrades. Therefore, assign features to tiers not only in terms of where a segment might land, but how a customer could become a power-user, growing in terms of functionality rather than changing their segment.

Q4: ARE ACQUISITION CHANNELS SATURATED?

With positive NRR, you can turn back to the question of new customers. If the rate of "new customers arriving per month" has stalled—or even sagged—you'll want to address that.

In fact, stalling and sagging is what marketing campaigns naturally do. I have covered this in detail, explained through realistic models and supported by real-world examples, in this article about exponential growth (p. 115) (Figure 12).

The first thing is to find the bottleneck of your conversion funnel, and try to widen it. Use data to determine what's restricting sales: Not enough new people coming to the website? Traffic quality is too low? Close rates too low? Very high churn in first three months?

Sometimes you can double growth by transforming the rate-limiting step. Careful though: Often "fixing" one step just sends the problem downstream, where it's even more expensive to detect and fix. For example: You increase traffic to the website, but it's low qual-

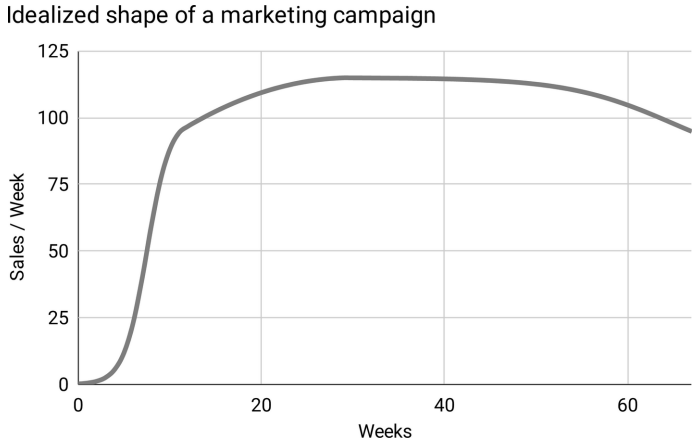


Figure 12

ity, so sales conversations decrease, so not only is growth still stagnant, you're also wasting more time in useless sales calls. Therefore, try to do the opposite, i.e. if sales conversion rates are poor, of course try to improve sales techniques, but also check whether you could increase lead quality upstream, where it's cheaper and more scalable.

Another thing you can do is accept a lower ROI. Of course you don't like over-paying for new customers, but in the long run it might not be a waste of money. This is especially true with positive NRR, because customers will grow over time, which means you can spend more up-front. This can create a cash-flow problem; annual plans (p. 353) can counteract that.

The next obvious thing to do is: Find more channels. If you have only one, this is almost surely the next step. If you're at scale, and already have multiple, it is worth experimenting (p. 1261) to find another, but it's more likely you have reached total saturation. There are a finite number of positive-ROI marketing channels.

More likely, you'll need to find a completely new technique for finding—or even creating—new customers.

For example, Hubspot hit a growth plateau when selling directly to customers (through multiple channels). The way they solved it was with a dramatically different channel: Reselling through agencies. It took years before that effort was making a material impact on growth rate, but then it became fully half of their total growth (Figure 13).

Quickbooks was similar; they sold their accounting software directly to small businesses through stores (before the Internet!), but most of the growth was through selling software to accountants, because the easiest workflow for the accountant was to import data from Quickbooks, which meant the accountants forced their clients to buy Quickbooks.

Constant Contact (email marketing for small businesses) hit a growth plateau, and it was expensive to advertise to the non-aggregated audience of "small businesses" who would pay only $20/mo. They realized that most small businesses didn't even know where to start with email marketing; that meant there might be 10x the market if

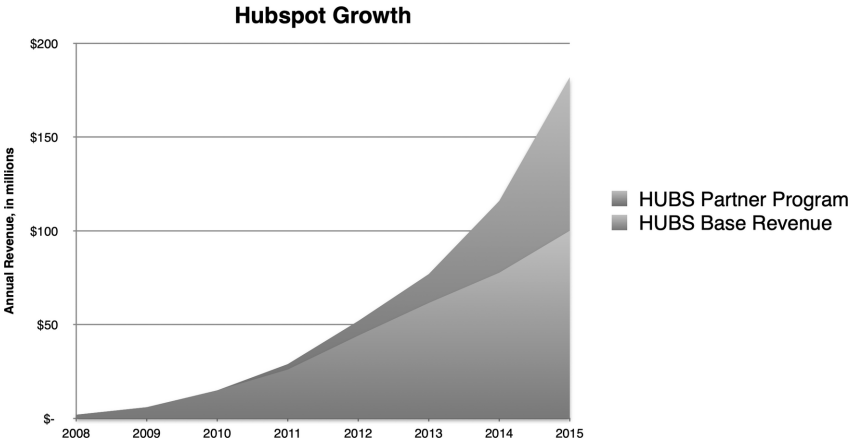
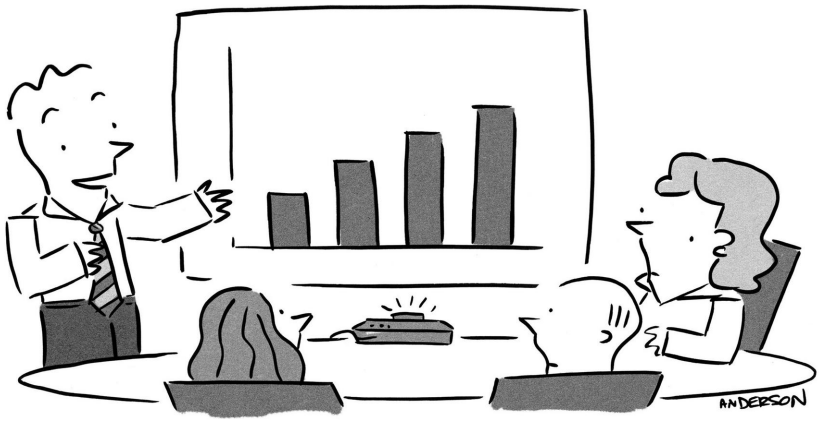


Figure 13

they could *educate* the business owners. Since those owners weren't internet-savvy, Constant Contact came to them—hosting myriad in-person training sessions, a few dozen business owners at a time, across dozens of cities in America. It worked; as expensive and non-scalable as that may sound, it broke through their stalled acquisition.

Q5: IS YOUR TARGET MARKET SATURATED?

Markets are not infinite, especially when you consider the small subset of the market for which your product is "perfect," i.e. the one housing your ICP (Ideal Customer Profile) (p. 317). Perhaps you are already reaching everyone who you can reach.



"What's really exciting is if you extend this out five years, we own around 400% of the market!"

credit 1123

Now it's time to expand the market. But in which direction? What is most lucrative? What is least risky? What makes sense for your company at this stage, at this time, with this budget?

See this article on expanding into adjacent markets (p. 793) for details on how to identify this and make the decision.

Q6: DO YOU EVEN NEED TO GROW REVENUE?

"If you're not growing, you're dying" as the saying goes.

Your knee-jerk reaction might be: That's what money-grubbing investors say! It doesn't apply to me! But, I've heard this phrase my whole career, including at bootstrapped companies where the founders hated VCs. The phrase is not necessarily wrong.

Still, it's not necessarily right. What *is* right?

You must decide what is important to you (p. 603). Maybe you want to maximize profit instead (but not at the expense of what's fair to customers or what you would be proud of doing). Maybe you want to minimize how much time you spend on work that you dislike. There are many more choices; the only wrong choice is to avoid making a clear choice, because then you can't aim the company at that choice.

For example, Alexandr Volodarsky[1124] hit a revenue plateau with lemon.io,[1125] then successfully pivoted to increasing profit instead (Figure 14).

Or, it might be time to sell (p. 45). Perhaps someone else has the energy and motivation to take it to the next stage. Maybe you simply want to do something else, now that you're older and have significantly more money in the bank and confidence in your head. Maybe

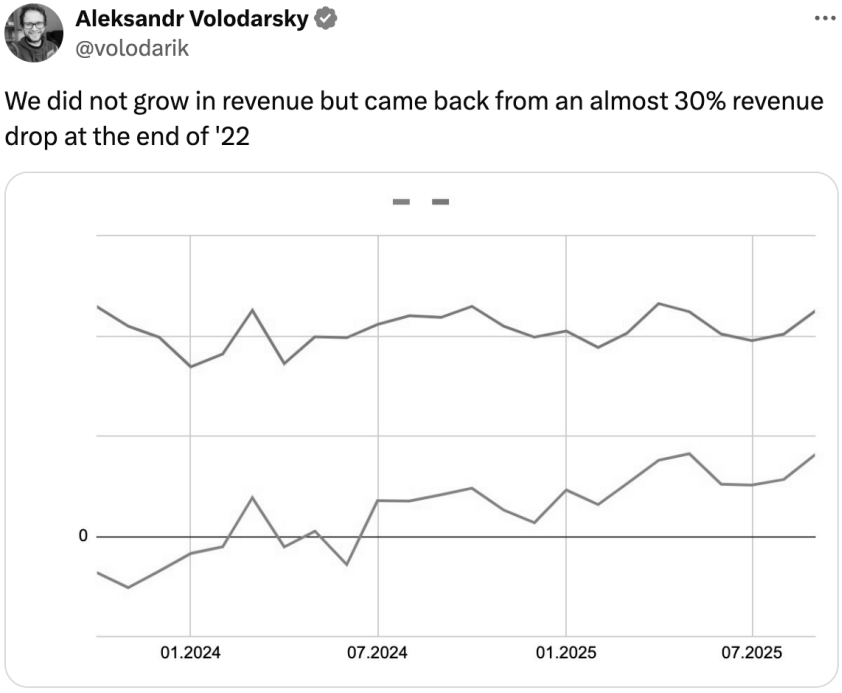


Figure 14: Revenue (red) remained flat while profit (blue) grew.

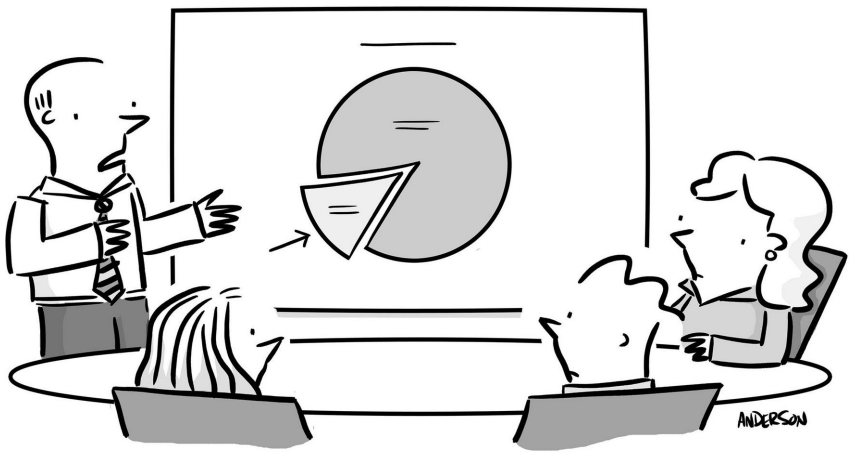
selling now maximizes money in your bank account, because if you wait another three years, revenue will continue to sag, and your sale price will be dramatically lower (both because of lower revenue, and much lower revenue-multiple).

No one can make that choice for you. Don't let others bully you either that you "must" sell or "must not" sell. It's highly personal. Listen to others, gather differing opinions, but treat them as "ideas," not "mandates." Maybe take a sabbatical and find out whether the prospect of returning to work fills you with excitement or dread. In that case, your "gut-feeling" is correct.

Congratulations on building a fantastic business! Now honestly diagnose what's holding you back.

Thanks to Josh Ho[1127] *for feedback on early drafts.*

Chapter 101:
Your target market isn't demographic



"Every year there are naughty children who don't get any toys. That's a vastly underserved niche I think we can capitalize on."

credit 1128

I don't like the traditional target market definition that focuses on demographics and firmographics:

- We sell to small businesses in the UK.
- Our TV show is targeted at females, 19-29 years old.
- Our fast-casual restaurant appeals to college students and young professionals.
- We target outdoor enthusiasts in the Pacific Northwest

Sometimes this is warranted. If you sell back-office software designed specifically for dentist offices under Norwegian law, then that description is accurate. But my experience is that, for most companies, this sort of market definition incorrectly categorizes the market.

For the TV show: Most females 19-29 don't watch it. Why not—what differentiates those who watch from those who don't? And many people of all types and ages *do* watch it; why? Getting to that deeper reason reveals the actual "target." Even if you're correct—that the named demographic includes many viewers—you're missing the point of your own show, and therefore targeting too many people (the females 19-29 who aren't actually your target) and also too few (all the others who are). You've avoided naming the actually "target."

Indeed, what will the advertisement for that show say? "Hey, if you're a female 19-29, watch this!" No, it will appeal to something else—people who like the genre, people who like those actors, people who enjoy that aesthetic, etc.. Whatever *that* is, is the target market, and should inform the advertisement.

How does this work for software? Consider an email client. It's something that people might use personally or they might use at work. It might be something used at a small company or a large company. It's probably not language-dependent. So these traditional market segmentation dimensions are irrelevant. It might have to do something else, like for people who get hundreds or thousands of emails a day and need special organization and management techniques and workflows. It may be for people who have assistants. It may for people who

get very little email. It may be for people who worry about security. It may be for people who do systematic outbound campaigns. It may be for a shared support desk. These attributes have to do with the use-cases of the customer, not demographic or firmographic dimensions.

Let's make this concrete with an apparently-inexplicable scenario that most B2B software companies will relate to.

WHY GOOGLE ALSO BUYS "SMB" SOFTWARE

In the early days of WP Engine, our website design (Figure 1) belied the fact that we were a small company, selling to the small end of the WordPress market. Our customers were SMBs (small businesses) or solo freelancers or very small agencies or individuals who had a successful blog. We were not positioned to be impressive to even mid-sized companies,[1129] to say nothing of large enterprises.

And yet, mid-sized and large companies bought too. Less in the first two years, but substantially in the next two years. But not big "Enterprise-sized" deals like multi-million-dollar contracts, but rather the same plans that the small customers bought. What did it mean? Should we "sell to the Enterprise?"

Many startups see this pattern. Twitter is full of founders with good, simple software, shocked to find that Google, Facebook, Salesforce, and Oracle have become customers, asking the melee-that-is-social-media whether this is evidence that they should "go up-market" with a sales team.

The short answer is: No.

The full answer is: You are accidentally and very understandably under the delusion that "company size" is a good way to characterize your target market. Sometimes that is right—for example if you're sell-

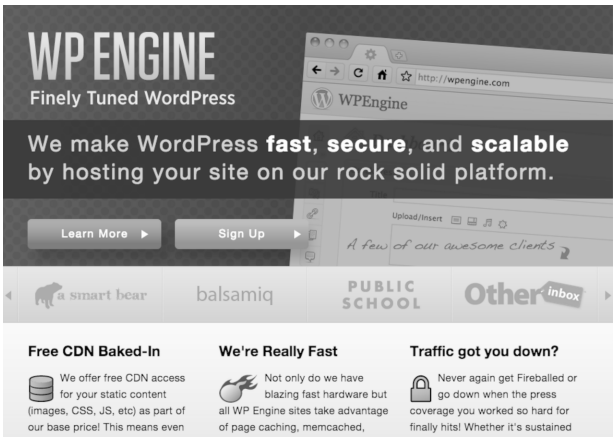


Figure 1: WP Engine's early website design screaming "We're an SMB selling to SMBs."

ing ERP software to large manufacturers—but actually it's *guaranteed* that software designed for small businesses will be purchased by big businesses.

The reason is that *individual teams* at large companies have needs too. Not everything at Google is purchased top-down and forced upon the roughly 27,000 engineers that work there. (If your software *is* purchased top-down and forced, *then* you are indeed selling to the enterprise, and that should be included in your target market definition.)

Those teams are buying for the same reason the only team at a small business is buying, and *that reason* defines your target market. (And should also define your advertising, positioning, pricing, and features.)

In my case at WP Engine, lots of people have WordPress sites, for lots of reasons. At a small business it might be their main website; at a large enterprise, it's probably part of a marketing campaign, or for a community project, or for a "labs" team, or all sorts of reasons:

- One of the largest banks in world used us for bank branches publishing their own content.

- One of the largest software companies used us for their developer-relations portal.
- One of the largest software tools companies used us for their public technical documentation.
- One of the one of the largest fast-food chains in the world used us to publish social content in small countries where they outsourced projects to local agencies who could get everything right (culture, language, color, etc).

That's because our target market wasn't "small businesses and individuals with WordPress." It was people who used WordPress who valued:

- Enterprise-*grade* quality (e.g. speed, scale, security, 24/7 high-quality tech support).
- A flexible, open-source, inexpensive-to-build platform where it was easy to hire employees or agencies anywhere in the world.
- Pricing that was 10x or 100x larger than "cheap hosting" but $\frac{1}{10}$ or $\frac{1}{100}$ as much as "enterprise CMS solutions".

You could continue asking "why?" Enterprise-grade, because they valued the website (as opposed to throw-away websites, or websites just for fun where price really is an issue). Flexible because they wanted to be able to change it themselves, or because they wanted to outsource anywhere in the world. Areas like consistent content-creation where WordPress is simply easier and better than other CMSs. Individual marketers who already knew WordPress from personal projects, and therefore could build and manage WordPress-based websites better than other CMSs.

The result of this chain-of-whys is internal clarity, and the path to effective marketing. For example, if your segment were "mid-sized companies," what would we say on the home page? "WordPress for mid-sized companies?" Great... soulless, boring, and what exactly the definition of "mid-sized" anyway? But now consider something

as simple as "Enterprise-Grade WordPress." Or perhaps "Affordable Enterprise-Grade WordPress." You could make it even more authentic (p. 627) and interesting and better, but at least it's being specific (p. 1439) about its strengths (i.e. affordable, reliable, high-quality). It's already an improvement.

HOW TO IDENTIFY YOUR ACTUAL TARGET MARKET

Techniques and frameworks

This is part of what you're doing when you find (p. 683) and then interview customers (p. 239). You're trying to get at their view of the problem you solve, how they describe it, what constraints they have, what would compel them to buy.

You use this information to identify your ideal customer profile (p. 317) and target everything for only them. Don't add company size, or industry, or geography, unless that is inextricably tied to the nature of the product, where it truly cannot be sold or used outside of that boundary.

You use the Needs Stack (p. 259) framework to define where you fit in into customers' "Maslow Hierarchy of Needs." You select the dimensions of Love and Utility (p. 275) at the intersection of your personal motivation and what will delight customers. These intersections are your target market.

In short, this isn't fundamentally different from existing techniques for understanding customers and markets. You're just seeing that there are more important characteristics than demographics and firmographics.

"We're transitioning away from small and medium businesses, and focusing instead on tall and grande."

credit[1130]

The testimonial test

The way you can check whether a certain customer profile [B] matches the target market of your existing customer profile [A] is by asking this question:

> Would customer [B] be swayed by a genuine, effusive testimonial given by [A]?

That is, would they say: Oh wow, yeah that person's situation is just like mine, so alright, I'm interested, I'll give it a try.

Sometimes proactively thinking about this question helps you to define the target market in the first place. It can even be used to directly test it—by seeing whether testimonials in sales calls light up people's eyes, or whether inserting them on landing pages increases conversions.

In particular: Gather a list of your best customers, where "best" is defined by high retention, propensity to upgrade, reports "10" NPS, proactively leaves positive reviews and advocates for you on social

media, engages with beta-testing or ideation for new features. Their shared characteristics help you define your target market, as would asking "what types of organizations and people would care about *their* testimonials?"

George Moore said it best in *Crossing the Chasm* back in 1991, when he defined "a market" as:

> - a set of actual or potential customers
> - for a given set of products or services
> - who have a common set of needs or wants, and
> - who reference each other when making a buying decision
>
> People intuitively understand every part of this definition except the last. Unfortunately, getting the last part—... the tendency of [a market] to reference each other when marking buying decisions—is absolutely key to successful high-tech marketing.

Notice that he, too, did not mention demographics* or firmographics, but rather a "common set of needs or wants" and that they see each other as references.

Market segmentation dimensions for the modern age

Historically, some market dimensions were forced upon us due to technical and societal constraints that no longer exist. It used to be that the only thing TV shows could know about an audience are things like gender and age, mediated by only a few firms like Nielsen, but that's

* Contrary to my formulation, he goes on to say that if two people buy the same product, even for the same reason, but don't actually communicate with each other, they are not in the same market. He gives the example of an oscilloscope for monitoring heart beats, saying that a doctor in Boston and one in Zaire have "no reasonable basis for communicating with each other," and therefore they are different markets. I believe this is less true in today's connected world (see the next section), and especially in software, I like the slightly less restrictive version that I gave above.

no longer the case. It used to be extremely difficult to accept credit card payments or transact in many countries and many currencies, complying with local taxes and trade laws, but companies like Stripe, PayPal, Paddle, and FastSpring have made that accessible to everyone, breaking down the traditional constraints of geography and currency. AI tech support and translations might soon break down language barriers.

We've already seen some examples of what dimensions we might use instead. Here's a list of ideas; most will not be applicable, but they'll get the thoughts flowing:

Dimension	**Notes**
Individual vs team	Sometimes can be "both," e.g. if a free single-player-mode gets people hooked, but teams is where the money is, e.g. project management like Asana or Notion.
Integrations	Nowadays, most software interacts with other software, whether part of a workflow, or because of the addition of AI, to-do system, project management, or chat.
Other software used daily	What else do they use every day, whether that suggests integration or just helps to characterize what customers expect and experience
Processes	What other processes and workflows do they use, or aspire to use, particularly as those intersect your product?
JTBD success	What objective numbers are they held to? Which are vital vs operational? Which are satisfied vs maximized?
Professional success	How are they evaluated? How do they get promoted? How are they seen as a success in their field? Early-career professionals have different priorities than mid-career or late-career individuals, regardless of company size.
Personal success	What fulfills them? What makes them happy? What is do compelling, they will put up with other things they don't like, or advocate for you within the company? (Their Needs Stack (p. 259).)

Dimension	Notes
Role inside company	Are they customer-facing or inward-facing? Do they interact with other departments or keep to themselves? What do others expect of them?
Title	What titles does this person with this role have? Probably a variety, especially if the product should span companies in different industries, or of different sizes.
Problem ownership	Who "owns" the problem you're solving? Is it centralized in one role or distributed across a team or organization? Is the problem-owner and solution-owner the same person?
Cultural attributes	Does their organization prioritize productivity or work-life balance? Do they teeter over ethical lines or stay firmly in or out of bounds? Are they focused on external innovation or internal efficiency?
Work style	Do they work from home, only in office, or hybrid? Are they separated from co-workers by time zones? How many hours do they work per day? Do they work synchronously or asynchronously?
Technical acumen	Where are they on the spectrum from "engineers who will argue with you about algorithms" to "technophobes reluctantly using devices they hate"?
Risk tolerance	Organizations analyze risk and accept risk differently. For example, creating new markets requires customers who accept risk. Risk-attitude is uncorrelated with company size.
Tech stack philosophy	Do they prefer open-source, proprietary, cloud-native, or on-premise solutions? Often stems from values, not just technical needs.
Update cadence	Do they prefer frequent incremental improvements or infrequent but substantial updates? Shapes product development rhythm.
Communication culture	Are they email-heavy, Slack-dependent, or meeting-oriented? Affects how they'll use and integrate your product.

Dimension	Notes
Organization size	SMB is often bought by larger, but not vice-versa. Best when selling top-down, or when the organization itself is relevant, e.g. HR software.
Organization growth trajectory	Is the customer in hyper-growth, steady state, or declining? Each stage creates different priorities and constraints.
Regulatory environment	How heavily regulated is their industry or function? What regulations? Affects feature needs, compliance requirements, and risk tolerance.
Budget type	Does the buyer have a fixed/annual budget, flexible budget, or "find money when needed" approach? Is there a clear budget threshold (p. 1239) where the sale is significantly different?
Sales process	Is this a top-down sale or bottom-up freemium / PLG? If the former, that might also be correlated with company size; the latter often isn't.
Decision cycle	Some make decisions in hours, others in weeks, others in quarters.
Emergency vs deliberate	Is the decision made because of an urgent condition (e.g. security breach) or is it made deliberately, with research, comparisons, or trials? Do they run pilots and A/B tests or make all-or-nothing commitments?
Adoption stage	Is the customer willing to take on technical- and vendor-risk in order to achieve special differentiation or exceptional results ("Early Adopter")? Does the customer want to see testimonials and other social proof before they buy ("Majority")? Is the customer being dragged kicking and screaming into the buying decision ("Laggard")?
Maker vs administrator	Is the primary user a person who is creating, making, delivering, or someone who is managing, monitoring, analyzing, reporting?

Dimension	Notes
Business model	Subscription, one-time, transaction-based, usage-based, freemium, ad-supported, etc? Can dictate what metrics they care about most, how they can pay, and how they see the product fitting into their value-chain or cost structure.
Learning style	Do they prefer video tutorials, documentation, 1:1 training, or figuring things out themselves? Shapes marketing, onboarding, and support.

Most of your answers will be "don't know" or "don't care," and you should invent new dimensions, but hopefully this gets you moving on the path to a better target market definition.

When in doubt, go back to your current, best customers. Your target market is the set of potential customers that your best customers self-identify with, in the sense that they would view each other's testimonials as compelling.

They themselves might not know exactly why; it's your job to figure out what characteristics define that set. But your best customers hold those answers. Through analyzing the problems they face, the way they wish to solve them, and the rest of their context, through observation and interrogation, you can develop a crisper list of characteristics that you can then target in marketing, identify in sales, and optimize for in product.

Discovering what you were always meant to be, and now becoming that, on purpose.

Chapter 102:
More or less

MORE FOR MORE · MORE FOR LESS
LESS FOR LESS

"I bequeath to you the fabled sword of Argoz. It grants the wielder great wisdom, untold strength, and a free gourmet cupcake on your birthday not to exceed five dollars."

credit 1131

There are three viable strategies in the interplay between value and price:

More for more
Get the best, with a price to match.
Luxury. Advantage.

More for less
Everything you actually need, at a reasonable price.
High-ROI. Practical.

Less for less
It's not much, but it's incredibly affordable.
A steal. Accessible.

You see this pattern across all industries, in both consumer and business markets.

More for more
Cardigan[1132] by Miu Miu for €2,150

In clothing, for example, a cashmere sweater embodies **More for more**—expensive, but deliciously soft (enjoyment) and warm (functional). A high-end brand has a function beyond high-quality materials: Status is bestowed upon the owner, due to it being overpriced beyond its quality. Buyers are not budget-constrained, which enables high prices and correspondingly high profit margins, but customers also have their pick of options, where every other product on the planet is a competitor. The customer is in control. Hence the importance of being "the best" according to some subset of the market.

More for less describes the practical decision-making that governs most purchases: Highly functional articles that will last for years, at one-fifth or even one-tenth the price of the same luxury item. Most people have a budget and therefore are seeking "The best that I can afford." Thus price is the primary constraint, inside which they prioritize features according to personal preference; they can't have everything, but they can prioritize everything. The product's goal is not to be the best along every dimension, but rather to offer a set of trade-offs that enough customers prefer over other sets.

More for less
Sweater[1133] by L.L. Bean for €92

Fast-fashion vendors provide **Less for less**, where you'll find the article disintegrated in the washing machine by the end of the season, but hey, it was cheaper than lunch. Consumers aren't choosing between trade-offs, but rather between having a sweater, or having no sweater at all. All negatives are accepted, as there is no alternative. The vendor enjoys a massive market, but rock-bottom prices mandate extreme cost-efficiency, requiring almost non-existent marketing and sales costs, and even in the best case yields barely any profit even at scale. A competitor with a better cost structure can decimate the business. Don't worry about it.[1135]

Less for Less
Hoodie[1134] by Tamu for €7

All three approaches are valid and, like all strategies (p. 489), each drags along a set of consequences, some of which you despise, others you love. You must pick one approach, and accept its batch of consequences. If you don't incorporate those consequences into the rest of your strategy, you'll have a confusing market message, a Frankenstein product, and the wrong pricing model.

MORE FOR MORE ("THE BEST")

Building "the best" is wonderful for both profit and enjoyment of life. It feels good to have pride in your craft, to have customers who love what you do and agree with your superlative self-assessment. "The best" commands a high price, which makes profit easier, even while self-funding further investment. This is a beautiful place to be, but it comes with consequences.

"The best" is not the average of a subset of things that already exist. The best is an outlier. The best is special. The best is different. Your ideas need to be substantially different from what's already available, or it's definitionally not true that you're the best.

So, the first question is: The best at what? It could be anything, but it has to be specific, and customers have to agree it's worth the higher price tag. It could be the best design—the simplest, the most pleasant to use, the most beautiful. It could be quality—the peace of mind and enjoyment of a product always "just works". It could be a superior workflow, could be that it integrates with all the tools the customer uses, it could be personal tech support, and more. This article on "Willingness to pay" (p. 275) lists more ways of being "the best".

Coupled with "best at what" is "best for whom?" There are billions of people on the planet who absolutely do not agree what "best" means in clothes, software, food, or articles about pricing strategy. This is where founders get fouled up; they think that just because *they* love it, and they "built it for myself" and therefore "I am the customer," and therefore other people will also to pay to solve that problem, in that way. They rarely do. The fallacies with that assumption (and some truths), are detailed here (p. 533). Instead, use your initial spark of insight as the starting point, your motivation, knowing that half of it is indeed correct but half isn't, and so you go find potential customers to interview (p. 683), and use a structured process to fix your hypotheses (p. 239) to morph your definition of "best" into alignment with what some specific ideal customer (p. 317).

Customers in this category care about something even more than they care about price. What kinds of things are like that? In business, an example is so-called "mission-critical" things, i.e. a failure or low quality causes outsized damage in internal cost, or far worse, damage to their reputation, in the worst case causing customers to leave.* Something that is an urgent, top priority for them right now, whether due to emergency or strategic mandate.** In the consumer world, it could be what Google calls a "toothbrush product"—something vital

---

* Our company WP Engine is an example, because we run websites like eCommerce (all of the customers' revenue) or media (merely being slow means they lose traffic, which means they lose ad revenue) or corporate home pages (downtime costs lost leads and lower reputation). This can also be CYA,[1136] i.e. "no one ever got fired for buying _____".

** See this article (p. 479) for many examples of how this arises, and how to define your target customer and product to match.

that is used at least twice a day, or ideally all day,* or tied to personal identity.**

Being "the best" isn't just about the product or the customer-fit; all other decisions must also be aligned with that positioning. Your marketing will focus on excellence and exclusivity rather than affordability. Your team will likely be smaller with higher-caliber individuals —passionate artisans who take pride in creating excellence and celebrate perfectionism. When hiring, you'll focus on finding the top 1% of talent, often paying premium salaries. Your customer support isn't measured in hours but minutes, often with white-glove, personalized service and dedicated account representatives. Your product development cycles may be slower, emphasizing perfection before launch, developing features without cutting corners regardless of cost. Because the number of customers in the market is small and limited, market-share matters, and upgrading existing customers is vital for growth at scale.***

Fortunately, the high price tag means you can invest in these things. And if you pull it off, it's the best for yourself because you're fulfilled when you're exercising your best qualities and you're validated when others agree on Twitter and say it to your face. You can recruit great employees, because who doesn't want to work at a company whose customers actually love them?

Most products are not "the best". But for those who can deliver it, it creates a virtuous cycle of excellence, satisfaction, and profitability that's hard to beat.

---

* Examples: to-do apps for busy people, email for people who spend more than an hour per day there, calendar apps for people whose calendars look like they only barely lost a game of Tetris, note-taking apps for people with 1000s of notes

** Like luxury brands, where being "far too expensive" is the point, like a ram with ridiculous horns.

*** Many public enterprise software companies get the *majority* of their revenue growth from up-selling existing customers, rather than adding new ones.

MORE FOR LESS
("HIGH ROI")

Efficiency is beautiful. "More for less" means receiving outsized value for a given cost. It is the right value proposition for most things in life, especially in business where you need to get a job done as cheaply as possible, but not cheaper.

The global apparel industry exemplifies the claim that this is the most popular category. The "more for less" category generates $1,200B of the industry's total $1,600B annual revenue,[1137] with luxury at $250B[1138] and fast-fashion at $150B.[1139] With profit margins of 5%-15%,* they are unsurprisingly more profitable than fast-fashion (under 5%)** and less than luxury (20%-30%).*** "More for less" software companies can have much better profit margins than apparel; coupled with reasonable prices and addressing the largest market segment, this is a great business model.

Which types of product benefit from high-efficiency, but doesn't require "the best?" Perhaps almost anything. Back-end systems that operate the company but are invisible to customer are a good start, because no competitive advantage is conveyed to the company for get-

* L.L. Bean, makers of the sweater above, is private and doesn't report profit, so this range is based on their 10% annual spend on bonuses and give-back activity[1140] and data from other retailers.

** The outlier-most-profitable fast-fashion brand stands at 9%, but almost all are under 5%.[1141]

*** For example Prada, which makes the Miu Miu sweater above, generates[1142] 23% net profit.

ting "the best." Even more ideally, areas where downtime isn't terribly disruptive to their real-time operations; a reporting system can be down for an hour without hurting their business, but a customer support tool cannot. Or an area where it's OK if not every desired feature is present, again like reporting software where the lack of exactly the right chart-type is not a deal-breaker, and won't prevent them from seeing the data. Even better when the product just needs to "tick a box" to "say that we have it," because then requirements won't be stringent. It might sound negative that your product isn't "critical to their success", but being non-real-time, non-essential, also means you have far more leniency in quality or lack of features.

This leniency in requirements is also why you will make 100x more sales than you expect, even though you still target the "perfect" ideal customer with marketing and product features. **For your ideal customer, you are in fact "the best,"**[1143] and they cannot believe they're getting the "More for more" product for the "More for less" price. That attitude defines them as the ideal customer (p. 1211); if you can't win *them* over from your home page, you're not doing your job.

But, for every one of those, there's a hundred others where you're the right choice. You have most of what they want, you don't have any deal-breakers, and the price is right, so you've made the sale. So, you gain the power of specificity, clarity, and emotional connection from talking to your ideal customer, while winning all the "More for less" customers anyway. This article on ICPs (p. 317) explains this dynamic in detail, with examples.

Although the main value here is economy, "save money" is weak positioning. Position by showing how much value you're generating rather than how much money you're saving as described in this article on positioning for value (p. 165). Because, even if this is your pricing strategy, that doesn't mean it's your marketing material. Your marketing position still needs to be as strong as possible, emphasizing that it really is *more* for less.

As with the other pricing strategies, the rest of your decisions must be aligned. Marketing will make the customer feel like this is the

"smart choice," neither wasting money on useless branding nor falling prey to the mistake of "If you think this is expensive, wait until you see how expensive 'cheap' is." Marketing must make the trade-offs clear, not try to claim it's the best for everyone, which quietly alienates everyone; comparison charts shouldn't be a solid column of green checkmarks, but rather elucidate where you are strong, and where you are not the right solution. Product features must not be too difficult to maintain, as the price tag doesn't allow for special exceptions or for brittle integrations that only 5% of the customers use. Rather, build only those features used by at least 50% of customers, or where at least 15% of customers buy you specifically for that reason. Hire team members who excel at identifying and eliminating waste without sacrificing core value. Your whole business must reflect the "More for less" mentality; your internal costs and decisions match your external promises.

Pundits have long eschewed the "More for less" strategy as a poor compromise between being either "the best" or "the cheapest." As Michael Porter famously wrote in *Competitive Strategy* (1980), there are only three strategies: (a) Differentiated Best; (b) Low-Cost Leader; (c) Niche Leader. Later thinkers pointed out that (a) and (c) are the same, differing only in market-scope. This leaves the Porter model with what I am calling "More for more" and "Less for less."

However, the customers have spoken, and in most consumer and business industries, it is in fact the "More for less" category that wins most of the customers and most of the revenue and most of the profits. We must not fall prey to the incorrect attitude that this is "a balance" (p. 589) between value and cost; we should instead recognize that "More for less" is a third, independent strategy, and leverage its strengths to the fullest.

LESS FOR LESS ("INCREDIBLE DEAL")

They say "never compete on price" (p. 437), but as the linked article demonstrates, that statement is incorrect. There are many wildly successful companies, from solo bootstrappers to some of the largest and most-profitable companies in the world,* who became successful due to absolute rock-bottom prices, even though those products come with severe trade-offs that many consumers would not accept. Sometimes, "cheapest" *is* "best".

It may seem like "less for less" couldn't be a good strategy, because neither employees nor consumers ought to be happy with "less," and low prices means low revenue, which in turns means little money for marketing, sales, development, support, product, design, or fun company-culture stuff around the office.** All that is true, and are among the consequences we accept when we adopt this strategy, but the hundreds of billions of dollars of profit (to say nothing of revenue) generated by the many "less for less" successes prove that these trade-offs are sometimes the right ones.

When something is surprisingly cheap, customers are not surprised that they have to compromise on most dimensions, such as quality, design, convenience, features, service, appearance, or longevity. Sure, it's missing some obvious features. Sure, it doesn't integrate with Enterprise systems. Sure, customer service is going to be a set of knowledge-based articles. Sure, it sometimes crashes. What did you expect for $5 per month?

There are billions of people who will buy $5 clothes that barely survive the season. If I pay $2/month for web hosting and I get email

---

* Examples: Amazon, Walmart, Vanguard, Costco, Dollar General, IKEA, Southwest Airlines, H&M, Zara, Xiaomi, ALDI

** Or whatever virtual-office-substitute we're pretending has the same effect as human beings being beings together.

and a website, it's okay if the email gets some spam and the website is not always fast or even always working. And I don't expect the clothing retailer to have customer service or accept returns, and I don't expect tech support from that web host to know anything beyond helping me reset my password.

The fundamental reason a customer will accept a wide range of trade-offs is that **the alternative is to not have the product at all.** That is, the choice is between having any hoodie versus having no hoodie. The product is "binary" in this sense—having versus not-having—thanks to an accessible price tag. One does not quibble over "features" or "quality" or "service" when the alternative is "nothing".

This is at once empowering—supplying goods to people who otherwise had no access—and depressing—that so many people cannot afford the comforts invented through modern ingenuity. And so we generate the worst versions of those items for most of the world to "enjoy." Which viewpoint is correct? The answer can often be found in the mission and purpose behind the company (p. 1169)—the real one, not the one written by the marketing department on the "About Us" page—where the fundamental driver of the company is either one of empowerment and impact, or one that exploits desperation. Regardless, the company should extol the positive aspects in branding and positioning: Emphasizing democratization, participation, simplicity, affordability, and accessibility.

The hardest part of executing this strategy is being profitable, since there's so little revenue to be profitable with. The costs have to be extraordinarily, unconventionally low. Some of that can come with scale, either directly (e.g. bulk-purchases from suppliers) or structurally (e.g. tech support per customer is lower at 100,000 customers than at

100, simply because most of them don't have a question in any given month). Still, it takes a lot of $2/mo customers before you can afford to hire even a single tech support person. Or if winning a customer means winning $2/mo and maybe $50 across the entire lifetime of that customer, you can't afford to pay very much to acquire that customer (p. 1377), whether that's in direct advertising spend, or sharing revenue with an influencer,* or indirect so-called "organic" marketing,** and you certainly can't have a sales team. You must rely on word-of-mouth and reviews from customers; you'll need to encourage it.

Given that revenue will significantly lag costs, is it mandatory that such a company raise money? The company will certainly consume cash for a while, whether that's in the form of someone's savings,

credit[1144]

"And for the connoisseur on a budget we have a lovely Tuesday vintage at a buck fifty."

* Affiliates are paid only after you are paid, which means this is an affordable form of marketing from the start.

** While harder to get customers at first (p. 1369), when pages aren't ranked in Google and social media accounts aren't followed by anyone, those assets do scale over time.

friends-and-family, a loan, or indeed raising money. Annual billing can help (p. 353) quite a bit. This is why pundits (including me) are always telling bootstrappers to "raise their prices" or, since you now know that prices are coupled with every other strategic decision in the company: "switch from a 'less for less' strategy to either 'more for less' or 'more for more.'" If you don't want to raise money, don't build a business that burns money on purpose.

However, it's not true that such business need to raise an exorbitant amount of money. In fact, in the same article above (p. 437), every one of the four case studies (which each grew to multiple-billion dollars in revenue) raised only a small amount of money. Those stories are detailed in the article. When the strategic decisions are fully aligned to a low-cost operation, you only need enough starter cash to get going. This is where VC-funded companies often get it wrong; they assume "we can work the costs out later," but when "later" comes, they realize their processes, technology, and culture has solidified around whatever worked for acquiring customers, and change is either impossible or dramatically painful, like layoffs, rewriting the code that works, and trying to force unnatural marketing and sales motions. There are also examples of bootstrapped companies using this strategy who never raised money; they typically have some extreme cost advantage, such as being located in a country with extremely low costs, yet selling to the United States and Europe. Being physically located in a low-cost zone is a competitive advantage that companies in The West cannot match.

As usual, what doesn't work is having a low price without the rest of the strategic decisions. This constantly happens with inexperienced founders who think: "My product sucks, so I'm not comfortable charging a lot."

That's not a good reason, and you shouldn't just build crappy product. Instead, the question should be: How do I build a simple yet lovable product (p. 101) that some specific ideal customer (p. 317) would pay 10x more than I'm currently charging, even though it

doesn't have a lot of features, and sometimes doesn't even work? That's exactly what I did with WP Engine (p. 9).

While the hardest intrinsic challenge is to be profitable, the hardest external challenge is that a competitor can beat you with lower prices alone. And this is all too easy, because there's so many reasons why a competitor might win:

1. Newer technology means they can built it for less.
2. Different location or other cost-structure.
3. Larger company uses a "loss-leader" to bury you, since they don't need it to make money at all.*

The antidote is to constantly increase your cost-efficiency, and possibly proactively lower prices. This is what Amazon did in retail and later did in the cloud.**

"Less for less" is brutal, because of relentless competition and having no extra money to work with. But, it means accessing the largest number of potential customers, and in the best case, you are truly enriching the world by bringing goods and services to people who otherwise could not afford to enjoy them.

All three pricing strategies are valid—each boasts wildly successful companies of all sizes, in every country.

---

* A particularly ironic example: Microsoft killed Netscape by putting over a thousand developers on Internet Explorer and releasing it for free; more than a decade later, Google killed Explorer by putting over a thousand developers on Chrome and releasing it for free.

** Amazon knew from their "your margin is my opportunity" strategy in retail, they should proactively reduce prices as their cloud operations scaled, to prevent people from looking for alternatives. Later, when Microsoft and Google were established as the only two serious competitors, they all stopped lowering prices simultaneously (p. 761). Surely a coincidence?

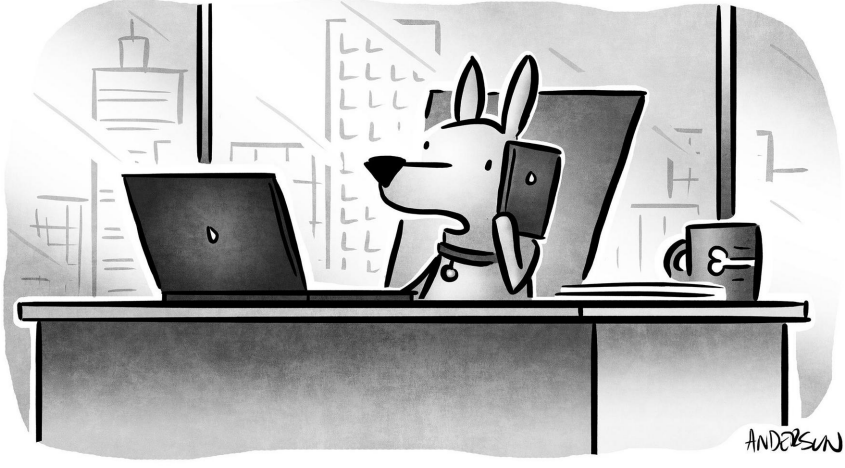
Ideally, one of these strategies jumps out at you as the smartest one, or at least the one you'd be the most excited and proud to implement. Another might jump out as being dumb, or at least too difficult, not worth the consequences, not where you want to spend your precious (p. 725), irreplaceable time on this Earth (p. 561).

Listen to your gut. Because all are, in fact, smart strategies, you should pick the one that best matches your personal strengths and proclivities (p. 569), because that's the one you will execute the best while being most fulfilled (p. 861). This is another choice that increases your chance of success (p. 667).

The only wrong choice is to not choose at all, and thus be confusing to customers and misaligned with the rest of your strategy. Then you will make internally-inconsistent choices, working against yourself, and your customers won't know how to think about your product, which means they won't be comfortable buying it.

And then your product is the right choice for no one.

Chapter 103:
When customers buy your competitor's product... and then buy yours



credit 1145

"Buy. Sell. Sell. Buy. Bark. Woof. Sorry. Buy."

Why would someone pay millions of dollars for software, and then also pay a bootstrapped company $700 for software that does the same thing?

How can you win those deals too?

"It's not a matter of life or death," most software developers say to excuse their bugs. But we were in a hospital. A modern hospital where everything is digitized, where it is not an exaggeration to say that if the systems failed, patients could die.

When Gerry and I walked into the hospital's network operations center, our faces lit up with an appropriate level of awe. On sales calls, the customer's jokes are always funny, and the customer's workspace fills you with awe.

Compared to all the other activity at the hospital, our product might appear insignificant. At ITWatchDogs our web-enabled hardware* measured indoor weather—the environment of server rooms, factory floors, or hospital rooms—things like temperature, humidity, whether the door was open, whether power was flowing, whether there was a problem with the air conditioning. If it gets too hot in a server room, components literally melt, causing downtime and significant replacement expense and the discovery that "backup" doesn't always include "restore". We were selling physical devices that monitored those factors and alerted when an air conditioner needs to be serviced ahead of an actual failure, or respond to a failure in minutes rather than hours.

The NOC (Network Operations Center) looks like NASA Mission Control—monitors everywhere, with numbers and graphs and kaleidoscopic colors that only long-tenured employees could pattern-match

---

* This was in the early 2000s, when "web-enabled hardware" was rare and interesting.

between "normal" and "concerning". Relatively few people monitor thousands of devices in the data center and strewn about the hospital—terminals and pagers and medical devices and patient records and the front desk and back-end systems, and who knows what else.

A hospital is the classic domain of the "Enterprise Software Buyer," and indeed they had "Enterprise Software" doing the monitoring. In particular they used HP OpenView—a multi-million dollar software extravaganza from Hewlett-Packard, in case you thought all that company knows how to do is charge too much for printer ink. Well, it turns out they charge too much for printer ink *and* they charge too much to monitor hospital IT systems. This is a company that can focus (p. 1119) on more than one thing at a time! We were hoping they would buy our weather gear, link it via HP OpenView to the Shuttle-Launch monitors, and save lives, or at least, not lose them due to IT failures.

But we noticed something else that surprised us. Something that turned into an instructive observation for anyone building a start-up and competing against incumbents (p. 295) including Big Enterprise Brands.

They were also using **What's Up Gold**.

You've never seen a more hideous logo.

Your first question should be: How could a hospital—that buys enterprise software—use a product with such a silly name? It wasn't because of the logo—that was even more embarrassing. Modern designers are sitting here, mouth agape, stunned. Why is it a ransom note? Why did the ransomer not have access to an apostrophe for "What's"? Does design not matter at all? (p. 853) This is an existential crisis; let's give the designers space to begin breathing normally again.

Your second question should be: What does that product do? And the answer is at once surprising and not surprising. It's not surprising because it monitors IT devices; hence "What's Up?" And when you

bought the more expensive version, you got "What's Up?"... Gold.* It was a simpler time.

It was surprising because they had bought and were also using HP OpenView. So why also buy What's Up Gold? What's Up Gold was a small bootstrapped company with shrink-wrapped software. How did they successfully sell a product when HP OpenView is sitting right there, at a cost of a million dollars?

So we asked them. This is already a lesson. You've got to proactively seek out potential customers (p. 683) and then talk to them (p. 239), to find out what's up. Whenever you see something surprising, that's a signal that there's something to learn. Dig in and learn it.

"Well," the guy said:

> "What's Up Gold has a couple of features we like. There's this thing we do many times per day, and it's really quick with What's Up Gold. And it's only like five hundred dollars or something. We can just run that on our credit card without asking permission. So we just bought it."

There's a couple of things to notice.

First, they bought two products to solve the same problem. We'll unpack that next.

Second, the pricing comment—it wasn't just that the pricing was *inexpensive*, it was that the pricing was *something that they could procure with neither effort nor consequence*. You probably think about pricing in terms of the psychology of the buyer ("Does $49.99 feel much cheaper than $50?"). But when you're selling to businesses—especially mid-sized to large businesses—it's less about the psychology of the buyer but rather how much of a pain in the ass it is for the buyer. When you understand pain-in-the-ass procurement (p. 737), you gain a new

---

* And you thought product names matter when selling to the enterprise? Maybe you also forgot our company was called ITWatchDogs and our product was the "Weather Goose." And a few years later I founded Smart Bear and sold almost exclusively to the enterprise (p. 751) with more animalia. Do what you know!

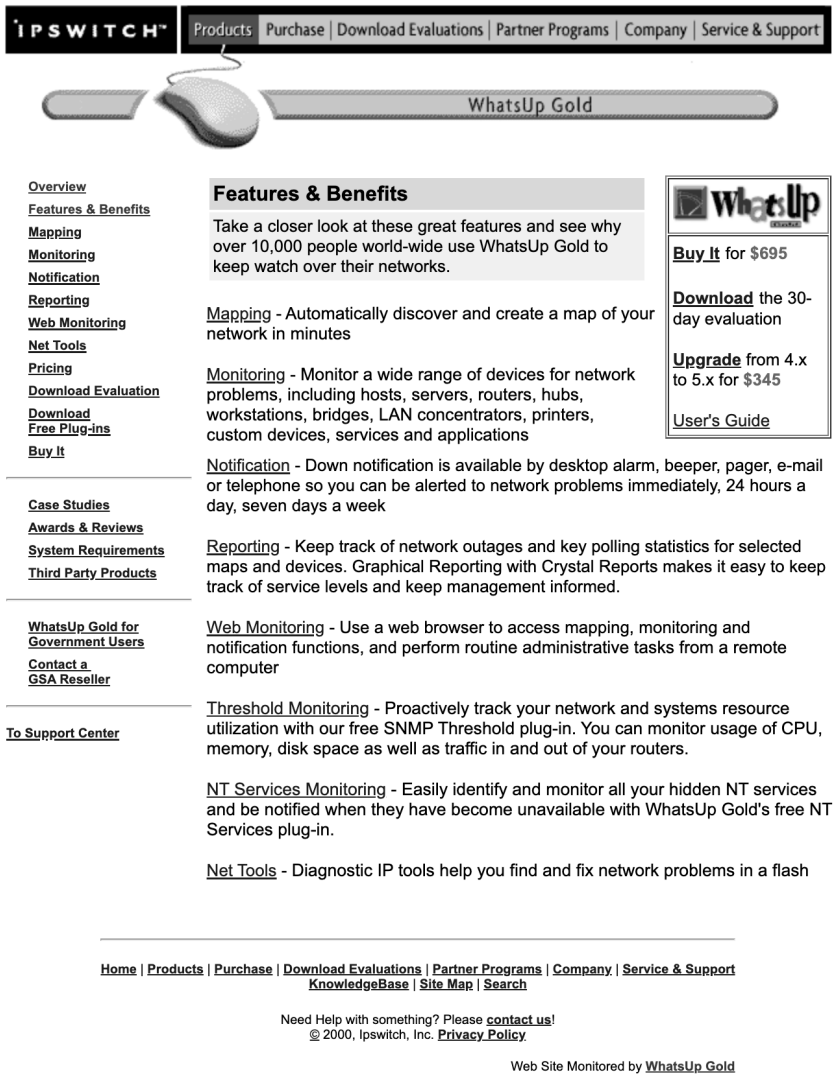
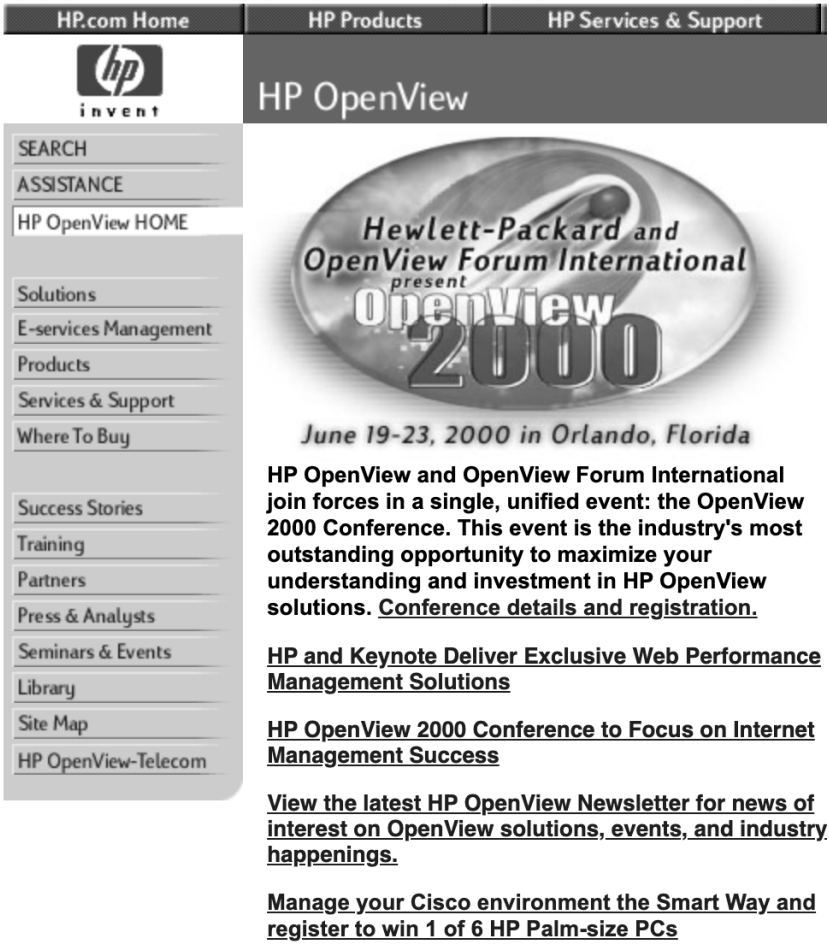


Figure 1: Can you believe this was the website for a product that costs $700? These were the days when website design truly didn't matter.

understanding of pricing. And you understand that $500 is not the same as $499.99; it's the same as $0.



credit[1147]

Figure 2: It's easy to forget how ugly the Internet used to be. HP was worth billions of dollars yet decided on this design.

It's like voting theory. Suppose there are four members of a board, with voting shares of 7%, 16%, 17%, and 60%. How much voting power does everyone have? The answer is: 0%, 0%, 0%, and 100%, because the last person always wins; the other percentages are irrelevant. Same with this customer: $500 is the same as $0, but $1,000 might be the same as $60,000.

And now you can see why they would buy a product to solve a problem that was (theoretically) already solved. Because from their perspective, What's Up Gold is free, so if it adds even a little incremental value, they'll buy it.*

Now consider this from What's Up Gold's perspective. Do they sell to the enterprise? Yes, in the sense that this customer is "an Enterprise". But they're not "selling to the Enterprise" in the sense that they were competing directly with HP OpenView. Other products competed directly with HP OpenView (e.g IBM's Tivoli, CA's Unicenter, BMC's Patrol—see, you are required to use acronyms (p. 751) and a "serious" product name to sell to The Enterprise!). What's Up Gold would never win against that line-up. But it doesn't matter—they sold their product to the same customer anyway. And now imagine that they could do that to every hospital—indeed every data center—in the world.

What's Up Gold thrived with:

1. A product with a few unique (p. 543) and delightful (p. 275) features.
2. A price that solved for procurement. (Notably *not* "price to value" nor perception tricks.)
3. A target market (p. 1211) defined by the needs of an individual IT person, not "SMB" vs "Enterprise."
4. A way to get the attention (p. 1027) of the right person (i.e. "distribution").

On that last point: We were curious how What's Up Gold managed to get in front of this IT person, especially since they'd already bought

* This is sadly not always the case; here's an in-depth discussion (p. 479), including the alternate cases.

a solution. So we asked about that too.* "Oh," the guy said, gesturing towards a table smothered in manuals, assorted wires, mangled Ethernet jacks, and loose CD-ROMs, "I might have seen it in one of those magazines, I don't know. I probably saw an ad in the back of one of the ones in the bathroom."

So while you're worried about SEO optimization and AdWords positioning, it's easy to overlook the magazine you take into the stall for 20 minutes of solace.

And yes, we later went to the bathroom to see which magazines were there. And then we advertised in them. And it worked. The things we do for success (p. 861).

Oddly, we learned a lot about the customer even though it wasn't questions about our product. We learned things when we asked about other products, as long as we were learning about the **life of the customer**. Maybe the customer was more honest than usual, exactly because it wasn't about us. We learned about a magazine we needed to advertise in. We learned customers would buy two pieces of software to solve the same problem.

And we learned about the credit card limit. So we changed the sales pitch on the fly. We came in expecting to pitch a $10,000 package. Instead, we said: "You could just buy $500 worth of stuff from us today. See whether you like it." Would they give it a try if it were free (p. 479)? Yes they would.

Each month they bought a little more. They were hooked—the product was good!—so they bought in a manner that was convenient for them.

---

* Sub-lesson: Always ask (p. 239) about adjacent products and behaviors. Wherever they found What's Up Gold, maybe they could have found us too.

It turned out that hospital was part of a larger hospital group. Eventually, we got orders for tens of thousands of dollars from their global IT center, because after it worked well in that one location, word spread inside the company, and then they really did go through procurement and make a big order. That's how Enterprise sales often works. It's not always a large top-down deal; it can start small, start accessible, start simple, start affordable, and then expand outward once you're a proven quantity. "Land and expand," goes the insipid but accurate saying.

That was 20 years ago, and SaaS software is different, and markets are different, and customer expectations are different, and budgets are different, and AI is different (p. 419), but what remains the same are the fundamental lessons of learning from your customers, and making things they want to buy.

Twice.

Chapter 104:
How many things should there be?
(Hint: Not 10)

How many things should there be on the list?

1: Focus
It's reductive to say only one thing is important (p. 603), but it also ensures that it gets done, as quickly as possible.

3: Magic Number
Humans across the globe have always liked when things come in threes. Also you can remember them all.

7: Because 7 ± 2
From one of the most-cited papers in psychology,[1149] seven items is what the average person can keep in their head simultaneously. If you can't keep it in your head, even when you're trying very hard, it's too many items.

credit[1148]

"Now I know that's a lot of acronyms, so here's an acrostic to help you remember..."

5: Because with post-COVID brains, now it's 5 ± 2

I have no data to support this claim, but I claim it nonetheless.

17: One per team

If you have 17 teams, perhaps you have 17 items. Normally this is too many items, but this is actually a repeat of "1: Focus" at the team level. It's still arguably too many if projects span multiple teams.

132: 100% completion

If there are 132 flags in the security audit, you could set a goal to address all of them. Getting through 100% of a specific set of things is a powerful statement. One could call this the same as "1: Focus" but with sub-items, where it's nice to have a checklist, and nice to feel progress as the team gets through the checklist, and clarity on why there are 132 items specifically, and a clear state-change when the very last item is finished (e.g. "We pass this audit, and then [large customer] renews for three years.")

402,302: Bankruptcy

An inbox with this many unread emails is the same as an inbox with no unread emails, except a number looms over your head. Declare "bankruptcy"—whether it's the email inbox or the Jira backlog. Archive* it all. The important things will come back.

Always Ask: **What is the purpose of this list?**

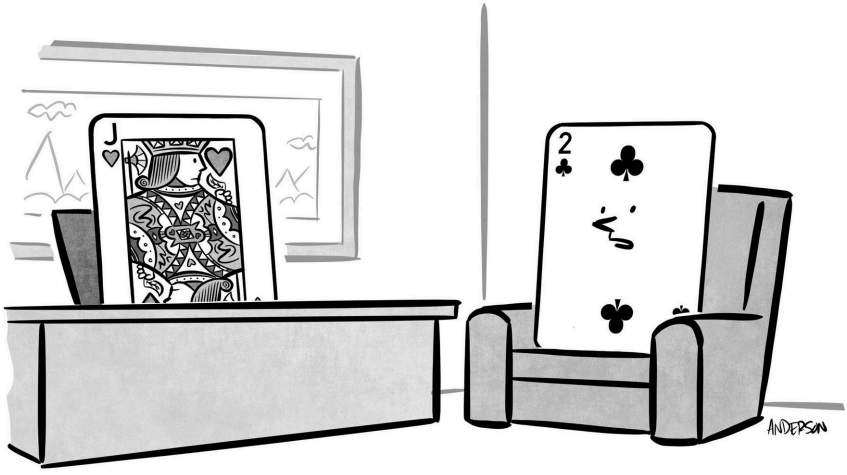
Given its purpose, should there be more things or fewer things? Is there a *good reason* why this the right number of things?

Usually, fewer is better, and the right answer is almost never "10."

* "Archive," rather than "delete," because disk space is close to free, and if you *really* needed something after all, you still have it. Not quite Marie Condo.

Chapter 105:
Metrics that cannot even be measured in retrospect

SINGLE FEATURE · CHURN · RISKS
MORE EXAMPLES



"Where do I see myself in five years?
Maybe a 5 or 6."

credit 1150

Some things cannot be measured, even after months pass and we have all the facts and data and analysis at our disposal.

If they cannot be measured in retrospect, they cannot be measured now, and you certainly cannot predict them or make a goal from them.

This sounds obvious, but many of these are so tempting, seemingly so important, we end up trying anyway. But it's a frustrating waste of time, and we ultimately fail.

Here's some common examples so you can train your pattern-matching engine, and see how to navigate the conversations.

IMPACT OF A SINGLE FEATURE ON THE REVENUE OF A PRODUCT

There are features at WP Engine which sales team pitch because people respond with genuine excitement. But then, after the sale, customers rarely use them.* Does that lack of usage mean the feature isn't contributing to revenue? No, it might have helped the sale.**

A feature that is used frequently is almost surely valuable; it's reasonable to claim that this feature is one of the main reasons why people pay for the product. But how much influence is it really? What if ten other features are also used frequently; do each "earn" 10% of the revenue? What if the feature is also used frequently by people who cancel—that would indicate the feature wasn't keeping them here after all. But maybe it was, and something totally unrelated caused the cancellation. What about all those other reasons customers are willing to pay (p. 275) that are unrelated to that oft-used feature?

---

* Don't worry, we don't charge extra for those features, so it's not unethical.

** P.S. Do you think it's wise for our competitors to copy those features? This is why you need to listen to customers instead of copying your competitors; who knows what's going on over there.

This is too difficult to unravel. This is why I like using a variety of KPIs (p. 645), only one of which is "usage." Of course product teams should celebrate when they create features that are widely used. It's just not the same thing as revenue.

Even after the fact, with 100s of 1000s of customers (in our case at WP Engine), it's still not possible to attribute revenue to individual features.

IMPACT OF INCREMENTAL ACTIVITIES ON CUSTOMER CHURN

As covered in another article (p. 71), customer churn severely limits growth and, more importantly, is a lagging indicator that customers don't want what you're selling. Therefore, it's almost always a good use of time to uncover the main drivers of churn, and address them with product, process, people, price, and other words that start with "p."

So the common and very reasonable question is: If we do X, how much will churn go down? And while we're actually doing X, how much is churn going down? And this leads to the crux: A year from now, will you *then* know the impact X had on churn? Unless X has an enormous and immediate impact, the answer is no.

The reasons often apply in cases besides churn, so let's list them separately to train your pattern-matching engine, while continuing to use churn as the example:

Cannot measure rare things, because signal is swamped by noise
If churn is 3%/mo, an initiative that reduces churn by 10%—a big impact!—will result in 2.7%/mo. How hard is it to measure a difference of 0.3%, month over month? How much does churn vary through pure randomness? Probably more than that. Some months have 15% more weekdays than subsequent months; if most customers churns on week-

days, that could make churn vary by 15% for that reason alone—again, pure noise. You probably just can't measure something that is so rare.

Lag between action and reaction
A customer who churns today has probably been unsuccessful for a while. They've been researching alternatives for weeks or months, weighing whether to pull the trigger. Larger customers might have tried to make it work for a year or more, then ran a multi-month process to investigate the options. So, an activity you start today is unlikely to change the trajectory of customers who have already decided to leave, and only today happened to push the red button.

Many causes of a result, means it's hard to measure a change in any one cause
There are many reasons why people leave. Some were never that interested, and didn't get hooked. Some got mad because of support interactions, even though they like the product. Some finally gave up waiting for you to make a feature they've needed, even though they love the support interactions. Some were never the right fit, but tried to make it work, and so did you. Some forgot they even had it unless their spouse asked what this credit card charge is. So any one action you take is likely incremental, difficult to measure in the fog of almost every customer *not* caring about that particular thing.

External forces that are larger than forces under your control
The market, competition, and the economy at large has a large yet unmeasurable impact. You can just go blaming the global economy—you have a business to run, and you are in control of more than you're currently controlling. Still, it's easy to map things like growth rates and churn rates onto the US stock market or inflation. Sometimes it can be the largest factor. How, in the face of these major factors, will you measure the tiny blips that your incremental work might accomplish?

Does this mean we shouldn't work on improving churn? Of course not—again it's one of the most valuable things any company can do, because retention compounds, and all the profit generated by a customer of a SaaS business happens in the later years, after they've paid back the sales and marketing expenses, gotten past the higher-than-average initial support load, and after your gross margins expand with scale.

But, trying to measure the impact of a single activity is folly;* even a year later, you'll never know how much that factored in relative to all the other things.

PROBABILITIES OF "RISKS"

The "Risk Slide" lists things that could derail the project or even tank the company. Executives quite rightly want to know: What is the chance that this will happen, and what is the magnitude of the impact if it does?

We can't invest in mitigating low-probability risks, although we might monitor the situation if the impact would be high. Conversely, high-probability risks might need proactive mitigation efforts. Thus the "probability of the event" is critical information, because it determines our actions.

But these probabilities aren't possible to measure, even after the fact.

Each risk is independent. It will happen, or it won't. You won't find out for months, maybe years. Then one or two happen and six don't. Maybe the one that happened was listed as a 30% probability,** and one that didn't was listed at 70%. Does that mean the probabilities were wrong? No, because 30% doesn't mean 0% and 70% doesn't mean 100%.

This is similar to weather forecasting. If there's a 30% chance of rain, and it rains, was the forecast correct? While you cannot answer that for a single day, there is an answer across hundreds of days: You

---

* The exception is when your churn is especially bad; anything over 3%/mo is scary. Then, sometimes it's possible to make large improvements.

** If you're lucky; I often see probabilities listed as "medium" or "low." Here's how to talk about probabilities (p. 997) properly.

credit[1151]

"But, to be fair, there's a fifty-percent chance of just about anything."

take all of the days for which the forecast was 30%, and see how often it did rain. If the real result is, say, 34%, the forecaster is doing a great job; if the real number is 70%, the forecaster needs improvement.*

That's great, but not for our Risk Slide. We don't have hundreds of measurements from forecasts of well-defined events. We have a slide of a handful of independent, complex risks, none of which have ever happened before. So we can't know whether our probabilities were correct, even years later when we know with complete certainty what actually happened.

So don't put probabilities on the slide at all. Only list the risks that you feel are so important that they either merit action or awareness.

* Actually that's not the full story; here is the fascinating math behind evaluating forecasts (p. 1319).

MORE EXAMPLES

Now that the general pattern is clear, here are more examples to avoid:

Measuring the effect of small design choices on user experience
Things like color, typography, layout, and word-choice, definitely matter, but typically noise overwhelms signal in attempting to measure it, especially in small iterations, and it's not effective to survey users about exactly which font they like better. Also, any UI change initially results in lower satisfaction scores, because people don't like change.

Measuring the effect or ROI of company culture or values
Having a strong culture, with a specific direction (p. 827), helps you create a healthy organization of people who are alike in the ways that matter, while creating diversity in all other ways. But, measuring the impact of a specific value or a specific culture-related initiative, doesn't make sense.

Measuring the value of brand
While there are some techniques that could make sense for the largest companies in the world (e.g. "unaided awareness" or "brand recall"[1152]), you generally cannot look at winning specific deals, or competitiveness, or growth, and answer "How much of that was due to brand?" This includes marketing campaigns that are primarily about brand, such as company-sponsored events.

Measuring the value of proactively fixing bugs or good engineering practices
You cannot measure the impact of a bug that never shipped, or the impact of an engineering practice that is simply how you work. There is no counter-factual.

Measuring "willingness-to-pay" or "value delivered"
I've detailed a theory of willingness-to-pay (p. 275) as a path both to profits and genuinely happy customers, yet directly measuring "value" or "willingness" is elusive. Even if you raise prices, and measure the change in the number of new customers or the ire and cancellation of existing customers, that still doesn't measure "value" or "willingness."

Anyway, that's a test you cannot run often, and a test that's often destructive for brand and customer relationships.

Measuring the efficacy of employee compensation or retention programs
Overall retention is easy and important to measure, but it is always multi-factored, and the rules can change over time. Thus this is like measuring the impact of a feature on product retention.

So stick with KPIs that measure specific aspects of your systems (p. 645), and stay clear of things you cannot measure even after the fact.

Chapter 106:
Stop saying “fail”

EXPLORING NEW TERRITORY · REVOLUTIONIZING
LEARNING · ITERATING · SOLVING PROBLEMS
MAJOR ERRORS

“Instead of calling it a failure, we just rebrand it as a dark and gritty reboot of previous profits.”

credit[1153]

It's not fail fast. It's not fail forward. It's not "fail" at all.

"Fail" is almost never the right word for what is happening. It's so final, so negative, and it's a dead end: What do you do after a failure? Mope?

Words have meaning. They can explain what's really going on, and suggest what should be done next. The word "fail" accomplishes neither.

EXPLORING NEW TERRITORY

If you're solving a maze, and a path doesn't work out, you haven't "failed," you've only backtracked. This is a necessary and natural aspect of exploration and creating businesses and products; in fact it's even called the idea maze.[1154] It often comes in the form of an experiment. The result of an experiment is evidence supporting or rejecting the hypothesis. Neither result is a "failure." A/B testing is a good practice, even though most of the time you will not have stumbled upon an improvement.

Words you could use instead of "fail" include: *experiment, test, bet, learning, investigate, trial, retry, probe*.

> *"There are two possible outcomes: if the result confirms the hypothesis, then you've made a measurement. If the result is contrary to the hypothesis, then you've made a discovery"*
>
> *—Enrico Fermi*

REVOLUTIONIZING

If you're creating something new, never accomplished by anyone before, and it doesn't immediately revolutionize the industry, you haven't "failed," you've just not yet found the breakthrough. Often innovations require many trials before the right solution is uncovered. If the right answer were obvious, it wouldn't be an innovation.

Words you could use instead of "fail" include: *backtrack, iterate, trial, attempt, step, miss*.

> *"I have not failed. I've just found 10,000 ways that won't work."*
>
> *–Thomas A. Edison*

LEARNING

If you're learning a new skill, and you stumble, get confused, get lost, or think you've mastered it only to find there's still a long road ahead, you haven't "failed," you're growing. No good teacher would get mad at a student for not learning a skill instantly, or even sliding backward. The learning curve is well-understood, and it is not a straight line.

Words you could use instead of "fail" include: *practice, cultivate, acquire, develop, learning curve, learning moment, trial, progression, evolve*.

> *I am always doing that which I can not do, in order that I may learn how to do it."*
>
> *–Vincent van Gogh*

ITERATING

Are you iterating on something that is fundamentally working, just making it better? This is optimization. Not all attempts at improvement will work, or have as much effect as hoped, but surely we shouldn't call ourselves "failures" when things are basically fine as they are.

Words you could use instead of "fail" include: *refine, calibrate, optimize, polish, hone, correct, fine-tune, dial-in*.

> *The only real mistake is the one from which we learn nothing."*
>
> *–Henry Ford*

SOLVING PROBLEMS

Are you solving a problem, but a minor one, particular to your current circumstance, where your high-level strategy and plan is still correct? Problem-solving in a complex space—whether a one-person startup or a thousand-person organization—cannot possibly proceed with in-

novation and speed and yet also no setbacks, no complications, no false-starts.

Words you could use instead of "fail" include: *iterate, adapt, adjust, course-correct, tweak, revise, morph, refactor, amend, fix*.

> *"Success is not final, failure is not fatal: It is the courage to continue that counts."*
>
> *—Winston Churchill*

MAJOR ERRORS

Have you made a major error, perhaps a complete misjudgment, incorrect strategic decision, or a project where months of effort by dozens of people has proved useless? That is indeed a big deal, and perhaps you're even justified in saying "we failed." Still, that doesn't explain its nature, and doesn't point towards a solution, which would be the constructive thing to do.

Words you could use instead of "fail" include: *Pivot, evolve, migrate, overhaul, revamp, transform, reconstruct, repurpose, reimagine, redesign, recreate, reinvent, reengineer, remodel, rebrand, realign*.

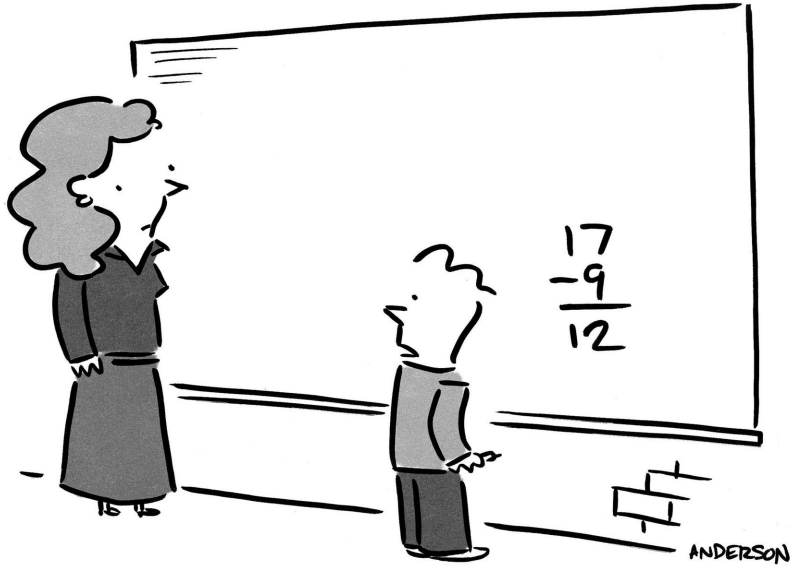
> *"Success is not built on success. It's built on failure. It's built on frustration. Sometimes it's built on catastrophe."*
>
> *—Sumner Redstone*

There are moments when we've truly, irrevocably, fatally failed, and we should say so. If a company has to shut down, especially before it achieved some meaningful success, then okay, we failed.

Otherwise, find a better way to describe what's happening, so we understand the situation better, and are pointed in the direction of what we should do next.

> *"Success is stumbling from failure to failure with no loss of enthusiasm."*
>
> *—Winston Churchill*

Chapter 107:
Who's lying?



"I know it's wrong, I'm just waiting for the autocorrect."

We had been flying for four hours, but both gas gauges still read "full." I didn't need a pilot's license to know that couldn't be right, nor to feel the rush of adrenaline in my gums at the thought of the engine sputtering to an eerie quiet death, propeller blades windmilling as we scream "mayday mayday" and "set it down over there" like in the movies, hopefully including the part where the heroes confidently stride away while the wreckage ignites in a fireball, such a banal event in their exhilarating life that they can't even be bothered to glance back at the carnage.

"Umm, this can't be right" I said to Gerry, the real pilot. "Yeah," he said, "the needles get stuck to the glass." He flicked the glass. The needle didn't move. "So... do we have enough gas?" "Yeah, we have another hour left, I stuck the tanks before we left."

"Sticking" means plumbing a wooden dowl through top of the wing, into the gas tank, judging the gas level by the height of the resulting wetness. Sometimes the simplest technology is best. Wooden sticks don't run out of batteries or make you wait forty-seven minutes for a security update.

> "*Trust, but verify."*
>
> *–Ronald Regan, repeating a Russian rhyming proverb: Доверяй, но проверяй*

It's not even good enough to just have "two of everything." If two things both rely on electricity, and the electricity goes out, you lose both. There were two gas gauges; both failed for the same reason. It needs to be differently-implemented as well, like a stick versus a gauge. For example, there's a normal magnetic ball compass floating in liquid that will work even if other power sources fail, but it's hard to read as it bounces around from the vibration so it's good to have another

one that runs off suction—air pressure differential between the outside and inside of the cabin—which is stable even when you're turning the plane in turbulence.

Gerry would say: **"Who's lying?"** Usually your instruments are correct, but sometimes one is lying. Maybe the suction system isn't working, so you double-checking suction-based dials with the electric-based dials. You stick the tank, in case the gas gauge is lying.

The same lesson applies to our daily life of data and metrics. You think you understand what each number means, and usually you're correct. But sometimes you're running out of gas and don't realize it. I've seen this happen for all sorts of reasons: The spreadsheet had a subtle bug in a formula, the analytics JavaScript code was accidentally left off one page, the survey email didn't get sent to all the customers in the cohort, the database query did/didn't filter something important, a nightly update script hasn't been running for three months.

A good way to check for bad data is to replicate the airplane dashboard method of deriving the same information in two different ways. Revenue from your billing system compared to cash flows from your bank statements. (Once I discovered our credit card processor was delaying our cash receipts.) Number of active customers from Stripe, and from your User Portal. (Because sometimes a cancellation in one system fails to cancel in another system.) Web traffic from Google Analytics but also another analytics system, or your raw web logs. (If you use five web analytics tools, they'll all give you different numbers; this *could* be due to differences in definitions of things like "visit" and "session," but is that truly all it is?)

Besides paranoia, I've found another advantage in computing the same data twice: a better understanding of the forces behind that data, and therefore better analysis of how the company operates and how the market is changing. Consider web traffic. There are analytics that tell you where traffic originates (imperfectly, especially with the latest browsers and extensions intentionally obscuring or blocking data), data about your advertising click-throughs, your own raw web server logs, and broad industry data (e.g. Google Trends on how search-traffic

for your keywords is changing). They all tell a different story. None has the full picture; all are biased. But taken together, your picture of the world is more complete, and biases might be cancelled through averaging or by paying heed only to the clearest, most consistent trends. If four different sources agree that a trend is happening, then it's definitely happening.

If a metric is important enough to watch it every day, and to act if its behavior deviates from expectation, then it's important enough to be double-checked. Both for accuracy, and for completeness of comprehension.

If your dashboard isn't redundant, you'll never know... who's lying?

Chapter 108: The Important Thing—powerful enough to override all your deficiencies

"I don't care *whose* time it is, we're going to have a nice dinner for once!"

credit 1156

Do you feel the crushing weight of the disadvantages facing every new company? No brand, no features, no customers, no money, no distribution, no search engine rankings, no efficient advertising, no incredible executive team, no NPS, no strategy.

How do the successful startups win anyway? Do they solve all those problems at once, or at least quickly?

No.

One answer is they pick their battles where they have advantages over incumbents (p. 295).

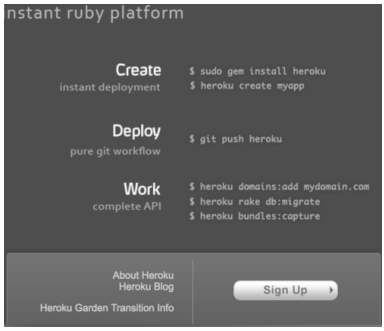
The other answer is that they need one thing to go really *really* right, to overcome the torrent of things that are going wrong.

credit 1157

I apologize in advance for using the dang iPhone as an example but... The iPhone is one of the most successful and important products of the past few decades. But the first version launched with a mountain of issues. It was a terrible phone, ironically. The whole idea was that it was a "smart *phone*," yet everyone agreed their cheap-o Nokia flip-phone was ten times better at being, you know, a *phone*. Also, imagine launching an operating system that didn't include "copy/paste." Terrible!

But, the iPhone did something so well, that people wanted so badly, they would put up with all the other crap: You could use the internet. The *real* Internet with full websites and everything. The web actually *worked* (even if slowly). Email actually worked. In your pocket.

It's hard to explain the magic and excitement to a Gen-Z'er who takes it for granted. This was so compelling, all the other problems didn't matter.



For more than ten years—an eon in tech—Heroku has been the dominate way that Ruby on Rails developers launch public applications. When it first came out, it was rife with "deal-breakers" that developers continually winged about. "What do you mean I have to use Bundler—it's broken half the time!" "What do you mean I can't change the filesystem at run-time—I'll have to change my algorithms!" "What do you mean it doesn't support MySQL—everyone uses MySQL! My queries are going to break." "Wow these websites are really slow." On and on with the complaints, and all quite valid.

But, Heroku did something so well, that people wanted to badly, they would put up with all the other crap. You could type `git push production` and your site went live. You could use a knob on a web page to determine how scalable the site was. (Don't worry, that knob is also connected to your wallet.) You never saw a server. You never thought about backups. You never worried how to securely stash your API keys. You always had a staging area to test things in a real server environment before pushing code live. DevOps became a thing of the past for a large class of applications. This was a revolution so important, so compelling, all the other problems didn't matter. Developers changed their workflows and their code around Heroku and "12-Factor"[1158] apps; Heroku did not change to suit developers.

This is a universal pattern I call **The Important Thing**.

Break-out products deliver something so fantastic, so game-changing, so important to the customer, that this one thing is sufficient to override the otherwise-overwhelming deficiencies of the rest of the product and company. So great that people tell their friends or force their colleagues to use it too (defeating the lack of marketing). So great that they'll use it even if support is slow and releases have bugs (defeating the lack of operational excellence). So great that they're excited to support a promising new company instead of worried about creating a dependency on a wobbly new company (defeating the lack of brand).

In short, despite the startup not having the positive attributes of large companies (brand, service, features, stability, integrations, social proof), the startup can still win (p. 295) because their Important Thing is so compelling.

The Important Thing isn't always a feature or technology. Fogbugz was never the leading bug-tracking system, but I employed it for most of the 2000s because I was a huge fan of Joel Spolsky's blog (p. 1553), so it felt good to use a product made by a company whose values and behaviors I respected and learned from. The same happened with Basecamp and the 37signals blog.[1159]*

Here's how to apply this to your own business:

It's easy to get overwhelmed by the myriad of inadequacies you undoubtedly have. It's tempting to attack them all, but worrying about everything and attacking simultaneously on all fronts with no weapons just leads to burn-out, and does not result in a company that is

* You might be tempted to say Basecamp was successful because the work-style espoused by 37signals leads to successful products, but the contrary evidence is that all of their many subsequent products—built with the same work-style, by the same people, and even with the same code base—were all dramatically less successful, to the point that all of them have now been discontinued, and the company has been renamed "Basecamp" to emphasize that only the first of those experiments was ultimately successful. *Editor's Note:* Now in 2024, they seem to be successful with hey.com.

excellent on any front. Fortunately, you don't need to solve all those problems. You need to solve almost none of them.

Instead, you need to focus on the one thing (maybe two) which is your Important Thing. The thing where, if you're extraordinarily good at it, customers will overlook everything else.

It could be a feature (e.g. disappearing messages with Snapchat), but you can look beyond features. It could be enabling a lifestyle (such as remote-work or with-kids-work). It could be your online reputation (e.g. Joel Spolsky for me). It could be that you're solving a problem in an industry that others overlook; having "any solution, even with problems" is better than having no solution. It could be that your culture resonates with an audience (e.g. 37signals), maybe due to an informal voice in an otherwise formal market, or because you have a cause—a higher purpose—so that people aren't just buying a shirt or some software but rather they are supporting a movement (e.g. Patagonia who cares so much about the environment that there's a company policy that they will bail employees out of jail if they're arrested for peaceful protest, and by the way one of the results is that they have only 4% annual employee turn-over).

You should select something that you *want* to obsess over for the next five to ten years, that gets your customers excited, and which you at least have a possibility of executing. And then do that. Maybe *only* that. If you let all the other fires burn, maybe you have a shot at actually being excellent at that Important Thing.

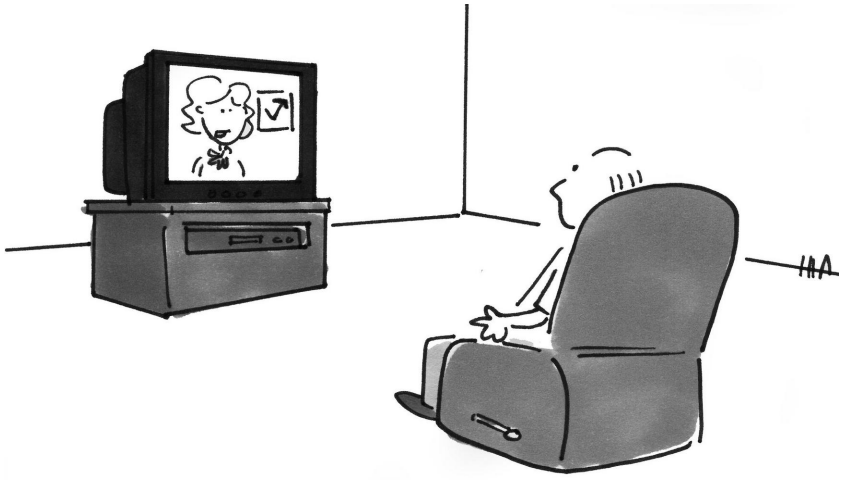
If you do, all those other disadvantages exist, but aren't fatal. That's all you can hope for, at the beginning (p. 429).

And all you need.

credit 1160

"Any way we can hurry this up?"

Chapter 109: Capturing luck with “or” instead of “and”



"Stocks rose slightly in early trading, then plummeted on news that stocks rose slightly in early trading."

I won a fake stock market competition in elementary school.

I put all my money in a few penny stocks—those are stocks whose prices are less than a dollar. Back then, stock prices could only be multiples of 1/8. Their small denomination meant that any fluctuation was a large (as a percentage) fluctuation in value. Some days I had the worst portfolio, other days I had the best. The competition happened to end on an up-day.

This was an example of "high risk, high reward." Like startups, but more random. (But are startups less random?)

Startups need luck too, in finding advertisement channels (p. 1369) that work, in the right mix of features (p. 1575) and usability that triggers product/market fit (p. 335), in cultivating a useful social media presence (p. 295), on employee number one (p. 1417) working out well, on a competitor[1162] not making a fatal move, on there being enough money in the market (p. 71), on appropriate pricing (p. 515), on market forces not shifting the rules of the game, and the list goes on. And that's after the luck of where and when you were born, the color of your skin, your gender, and that list goes on too.

When you put it that way, it's obvious why startups fail so frequently! They need a lot of success in a lot of areas, which is a lot of "good luck" (p. 1035) to string together.

How can you reduce this effect, and maybe even turn luck to work in your favor?

The list above is a bunch of "and's." That is, you need a good marketing channel *and* you need a few killer features *and* you need great initial employees *and* you need a healthy market, etc.. **"And" is bad!** It's bad because each one has a probability of success, and you compute the total probability of success by multiplying them.

No matter how optimistic you are about those probabilities, the end product is a small number. 70% multiplied by itself six times is only 12%; most of these things we're listing have a probability less than 70%, and there are more than six things that have to go right.

You can increase the chance that some of those things go right by making explicit choices. For example, you can pick a large, grow-

"Serendipity is up, fluke is doing well, but I'm a little concerned about our dumb luck."

credit[1163]

ing, and healthy market (p. 71) so that the risk that "no one wants to solve this problem" goes to zero. You can compete in a niche where incumbents don't care (p. 295) or cannot move quickly. You can hire someone you've worked with before, or build something sustainable without hiring. You can generate Love (p. 275) rather than only Utility or Coercion to overcome your deficiencies.

Here's a long list of those things and how to mitigate them (p. 379).

Even so, there will be plenty of challenges, so we need a second technique for boosting probability: **Leverage "or" instead of "and."**

Consider marketing channels. You could get your first few hundred customers through GoogleAds, or Facebook ads, or affiliate sales, or targeted outbound sales, or SEO, or guesting on podcasts and newsletters, or partnering with a high-profile reseller, or great press about your unique brand and message. Only one of these needs to work! So although the probability of success for each one of these is low, the probability that *something* will work is higher.

This is true of everything from product features to website copy for conversions to avenues to exiting the company years from now. The general rule is **optionality is strength.** When there are lots of

ways for things to go right, that is a strong position even if you haven't actualized one of those ways.

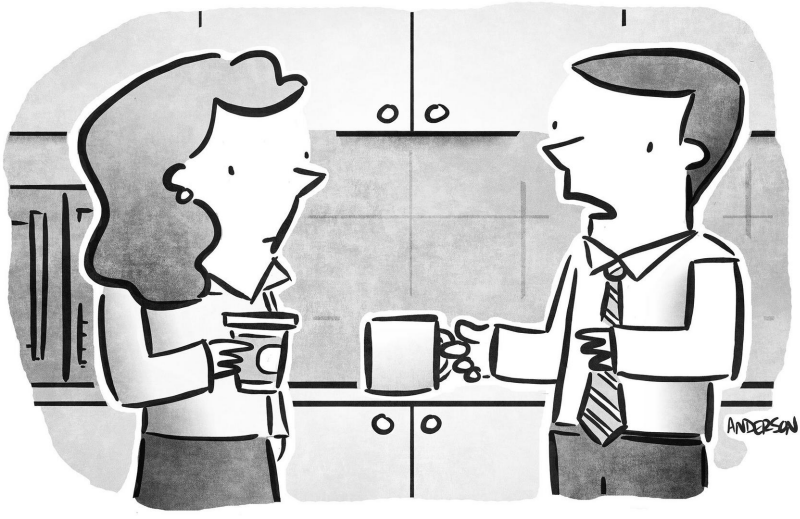
The converse of this is a business that has extra "and" clauses—even more than usual. Marketplaces, for example, almost never succeed.[1164] When they do succeed, they are often durable and profitable, which makes them a smart bet for a Venture Capitalist that can maintain a diversified portfolio of attempts, but for the individual business it's a tough road (p. 867). A marketplace has to thrive both with the sellers *and* the buyers, *and* it can't be too expensive to build up both sides, *and* both sides need to realize value, *and* it must not be easier/cheaper/preferable to transact outside the marketplace, … otherwise the marketplace fails. Also, many marketplaces often only deliver value once they are at scale; so another "and" is that they have to also "scale down" so the first 100 buyers and sellers also see value.

By accumulating "and" requirements, you are lowering the probability of success. By stringing together possible solutions with "or," you are increasing the number of ways that luck could smile upon you.

Set yourself up for luck!

Chapter 110:
Distributed Logical Time

OBJECTIVES & DEFINITIONS · SINGLE REPLICA
"HAPPENED-BEFORE" · SKEW · SIMULATIONS
PROBLEMS · ANTI-OBJECTIVES



"We're behind schedule on the time machine, but we'll catch up once it's finished."

credit[1165]

Time is notoriously difficult to synchronize in distributed systems. Many algorithms require—or have better characteristics when—any number of independent replicas achieve a single global time-ordering. In particular, it is useful to know that if two events are separated by more than a few seconds in real-time, that they will be correctly ordered, despite replicas having unsynchronized physical clocks.

> *"We're obviously in no danger of arriving at consensus."*
>
> *—Warren Buffett*

However, operating system clocks are undependable and can even move backward. Furthermore, we don't want to rely on a "master" or central server. Replicas should be able to arbitrarily join or leave a collective without announcement, operate independently, and communicate with each other peer-to-peer, in any pattern.

Common methods to solve the problem either involve custom infrastructure to achieve tight bounds on the behavior of operating system clocks (e.g. Google TrueTime[1166]), or some sort of distributed logical clock such as Vector Clocks.[1167] The former is unavailable to most of us, especially if we want our code to run in browsers or laptops or non-Google data centers. The latter suffers from unbounded memory requirements and still results in "conflicts" which are resolved arbitrarily or not at all.

Here we present an algorithm and Golang implementation that meets the objectives, with no assumptions about the behavior of operating system clocks, without vector clocks, and without central coordination.

OBJECTIVES & DEFINITIONS

Definitions

It's useful to establish a few definitions and conventions before describing how we achieve a solution:

- **RT**—Real Time—theoretical omniscient "actual time". Imagine it as arbitrarily-high-precision UTC time exactly synchronized everywhere. This is not a quantity that any replica can compute; it is for algorithmic discussion only.
- **PT**—Physical Time—the time reported by the operating system. This is the only notion of "time" that a replica can access.
- **LT**—Logical Time—our synthetic implementation of "time", with the properties suggested above and precisely defined below.
- `XT{expression}` means "the type of time `XT`, applied to the replicas or events in the expression." For example, `PT{ P<Q }` means "the Physical Time on the replica `P` is less than the Physical Time on replica `Q`."

Requirements & Assumptions

With this terminology, we can define our assumptions and our requirements for LT precisely:

1. **PT is arbitrary.**
   It can move backwards, it does not produce unique values, it might update only rarely, it is never synchronized between two replicas, it can be ahead of or behind RT by an arbitrary and variable amount. The only thing we know is that, over a long duration, it generally increases.

2. **LT always increases.**
   Whenever LT is requested, it will have increased since the previous request (regardless of PT's behavior).
3. If B happens after A in real-time, i.e. `RT{ B>A }`, but A and B happen on different replicas:
   a. **Correctly ordered in LT when sufficiently time-separated in RT.**
      If `RT{ B-A } > e`, where `e` is a small, bounded constant, then `LT{ B>A }` also.
   b. **Arbitrary LT order if close together in RT.**
      If `RT{ B-A } < e`, then there's best-effort for `LT{ B>A }`, but it could be `LT{ B<A }`.
   c. `e` **is small and bounded.**
      `e` must be small (e.g. 1-2 seconds), and be a constant, not proportional to some state or configuration. `e` is the time window inside which we accept events that are incorrectly ordered, so we require this window to be small.
   d. **"Happened-before" relation is always correctly ordered in LT.**
      If A happens, then the two replicas communicate, then B happens, then it is *always* true that `LT{ B>A }`, even if `RT{ B-A } < e`.
4. **Skewed PT on one replica doesn't skew the behavior of the collective.**
   If one replica's PT differs significantly from RT, it should still participate properly with the collective; for example it shouldn't automatically "win" due to the late-skewed PT, or be unable to write a change due to early-skewed PT.
5. **LT uses constant memory.**
   Use a fixed number of bits to represent LT, regardless of factors like the number of replicas. (Unlike Vector Clocks.)
6. **Decentralized, without "join" or "leave" events.**
   No leaders, no special designations, no central API. Peers can

communicate in arbitrary patterns. Peers never need to announce their joining or leaving the collective. There is no list of peers.

It may sound impossible to achieve the goals of LT given that RT is inaccessible, PT lacks all the properties we need, and replicas aren't synchronized. It's fun to see that in fact it can be achieved.

LT INSIDE A SINGLE REPLICA

It might seem trivial to create LT within a single replica, but even that requires an algorithm. Typically the operating system is used for PT, but PT can move backwards (e.g. with NTP, with leap-seconds, with manual settings), and is non-unique (e.g. asking for the time in rapid succession often yields the same value from the operating system, due to CPUs being faster than the precision of time), and can drift significantly far away from RT (due to various causes as shown in these examples[1168]).

PT will generally increase in the long-run, but we have to "smooth over" the bumps where it is stagnant or decreasing. To do that, we store the "last-seen PT" as well as a counter, and use the following algorithm (plus a mutex for concurrent access):

```
func GetLT() {
        if PT_Current > PT_LastSeen {
                PT_LastSeen = PT_Current
                Counter = 0
        } else {
                Counter = Counter + 1
        }
        return {PT_LastSeen, Counter}
}
```

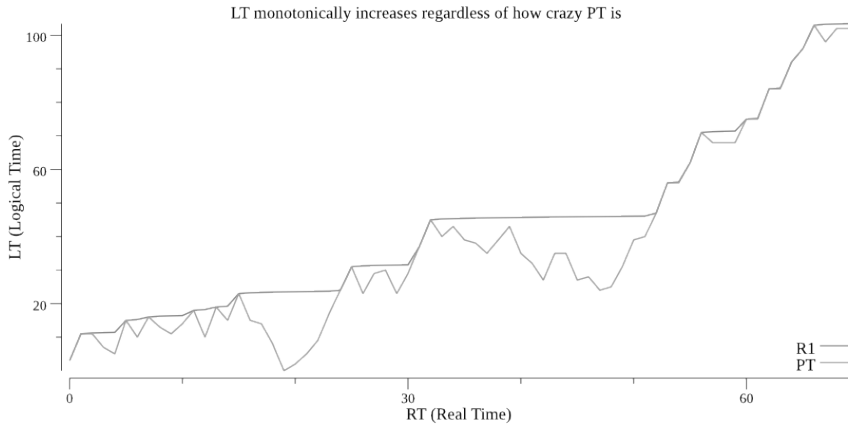


Figure 1

LT is ordered first by `PT_LastSeen` and then by `Counter`. So, as long as PT increases, we use it, staying hopefully somewhere near RT (more on this later). But, if PT does not increase, the Counter ensures that LT still increases.

In the following simulation from our golang implementation, we can see how LT monotonically increases even when PT doesn't (Figure 1).

The use of a counter also means the precision of PT is irrelevant. Therefore, a performance optimization is to use a separate recurring timer to update a thread-safe global variable with PT 1-4 times per second, rather than invoking the much more expensive and blocking operating system call to retrieve PT every time we compute LT. With this optimization, we achieved tens of millions of invocations per second even in a Javascript implementation on a typical laptop.

"HAPPENED-BEFORE" RELATION BETWEEN REPLICAS

Suppose an event A happens on a replica P, and then P communicates with another replica Q, sending that event. Afterwards, an event B happens on Q. We want to be certain of `LT{ B>A }`, even though A's LT was generated on P and B's LT was generated on Q (Shown in Figure 2).

The PT of P and Q will differ, and could differ in either direction. If `PT{ Q>P }`, then we'll get `LT{ B>A }` naturally, because the PT component of Q is already ahead of the PT component of P. That's the easy case.

In the other case that `PT{ P>Q }`, we have a problem. In the example below, P's PT is one minute ahead of Q's. After P sends Q event A with `LT{ 71.0 }`, Q's LT is still far behind, which means when event B happens one second later, it is `LT{ 13.0 }`, resulting in the problem `LT{ A>B }` even though `RT{ A<B }` (Figure 2).

To fix this, we simply set both components of Q's LT equal to P's. Q will know to do this, because when P communicates with Q, it transmits its current value of LT. The following algorithm ensures Q will end up with a strictly-larger LT:

```
func UpdateLTFromPeer( LT_Peer ) {
        // Operate on current LT
        LT_Local = GetLT()
        // Take the latest LT
        if LT_Peer > LT_Local {
                LT_Local = LT_Peer
        }
        // Ensure strictly larger than any previous LT
        Counter = Counter + 1
        // This is the new local LT
        SaveState(LT_Local)
}
```

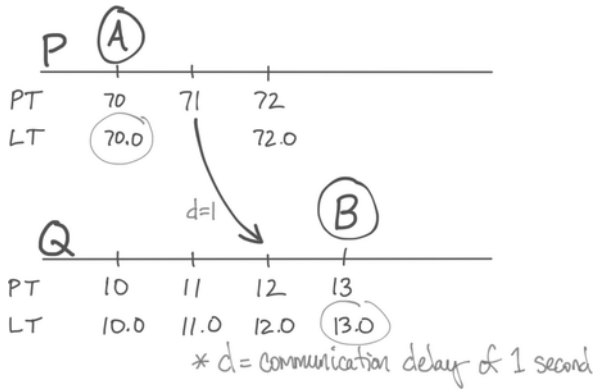


Figure 2: Problem: B's LT isn't later than A's LT, because they're on different servers.

A side effect is that Q's `PT_LastSeen` will be ahead of its own PT, but that's fine because Q will just use its Counter until its own PT catches up. Meanwhile, `PT{ B>A }` is guaranteed, as the diagram now shows (Figure 3).

Further discussion and a proof of correctness can be found in the paper that invented this method,[1169] in which it is called a Hybrid Logical Clock (HLC).

USING SKEW TO FIX THE FUTURE

Although the algorithm above satisfies many of the requirements of LT, it violates the requirement that `e` be small and bounded.

To see why, let's extend our example to consider what happens with subsequent events on P and Q. In particular, P generates an event C soon after B (in RT), and then Q generates an event D about thirty seconds after that (Figure 4).

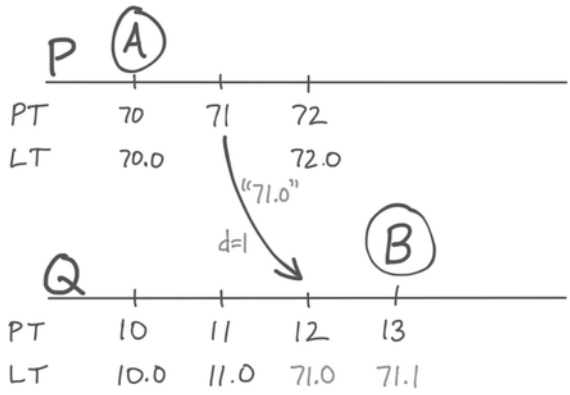


Figure 3: Solution: Q appropriates P's LT, because `LT{ P>Q }`

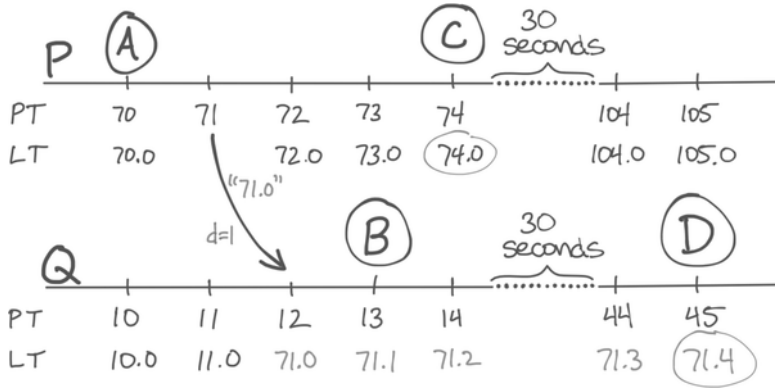


Figure 4: Problem: PT clock-skew breaks LT ordering of future events C and D, even though they differ by 30 seconds in RT

We've highlighted the problem: `LT{ C>D }`, even though `RT{ C<D }`. The cause of the problem is displayed in the diagram: Because

Q's PT is so far behind P's, Q has to use its `Counter` to increment its LT, but meanwhile P is incrementing its LT using its PT. In fact, *every* event on P during the minute after P communicated with Q will have an LT greater than *every* event on Q during the same interval, regardless of their ordering in RT. This is the condition described in our original LT goals where events can be mis-ordered in LT if they happen closer than a duration `e`. The trouble is, `e` is too big (it's one minute in this example) and it's not bounded (it could just as easily be one hour).

This situation remains even after Q's PT catches up with the synchronization event. P's LTs will *always* have a larger PT component than Q's LTs, and thus P's events (within any one-minute time window) will always look like they are later than Q's in LT, regardless of their order in RT.

One solution is to mandate small PT clock skews, which in turn mandates that `e` is small. For example, a basic NTP service can keep clocks synchronized to within tens of milliseconds. In this case the effect in our example would still exist, but only inside a tiny time window of tens of milliseconds, not a full minute. That would be acceptable.

However, our implementation does not assume control over PT. A replica might be a browser or laptop that we don't control, or a virtual machine that isn't running NTP. So, we need an algorithmic extension that eliminates this problem of PT clock skew.

The solution is for Q to compute the PT clock skew, and use that as an offset to its native PT to stay reasonably up-to-date with P. This cannot be done precisely, because of the non-measurable and often-variable communication transmission delay between P and Q, and because PT isn't dependable, but it turns out being precise is not necessary.

When P communicates its LT to Q, Q computes `s = PT{ P-Q }` as the "skew." As the diagram below illustrates, `s` will always underestimate the actual clock skew, because it's not taking transmission-

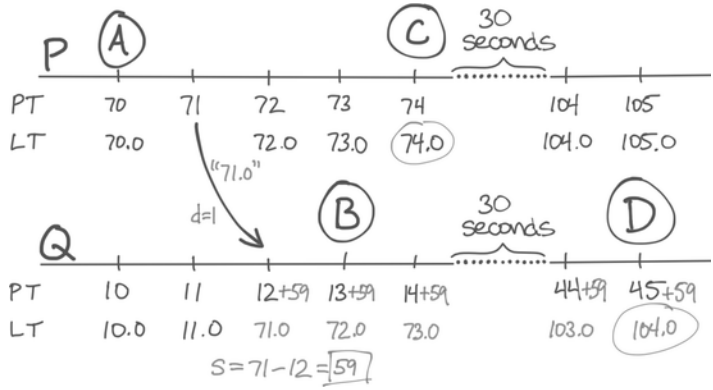


Figure 5: Solution: Use an approximation of real clock skew to reduce the time-window of out-of-order LTs

time `d` into account. In our example here, the actual clock-skew is `60`, but the computed skew is `59` due to the transmission delay `d=1`.

When `s` is positive, Q saves `s`. The next time PT is computed, Q uses `PT{ Q } + s` as physical time. This means Q's idea of physical time is now only `d` behind P. This nullifies the problem in our example (Figure 5).

If `s` is negative, it is ignored; this ensures that clocks that are already ahead do not get even further ahead.

Although in practice `d` is not measurable and fluctuates, it is always non-zero, and rarely larger than a few seconds. It is proportional to network transmission time, not proportional to PT clock skew or any other system state or configuration. Therefore we can say that `s` always under-estimates skew, and by a bounded amount on the order of the replica's communications delay (i.e. 1ms inside a data center, 100ms across a country, or 1000ms across the world).

Finally, observe from the diagram that the time-window in which this problem can occur has been reduced to just 1 second, i.e. reduced to `d`. Indeed, `e` from our LT goals is exactly `d` theoretically, and on the

order of d practically, which we just said was less than a few seconds. Thus, we have achieved the objective that e be a bounded constant.

What happens when all replicas' PTs are in fact well-synchronized, e.g. with an error less than d, which is easily achievable with well-known algorithms like NTP, or modern phones and laptops that synchronize their clock with GPS? Then the computed skew will be less than zero. To see why, consider that `[computed skew] = [real skew] - d`, but in this hypothetical, `[real skew]` might be 50ms whereas d is typically greater than that. Negative computed skews are ignored, thus we'll always have `s = 0`.

Although this is not a specific requirement on the behavior of LT, it does satisfy an intuitive desire for skew-correction to vanish when it isn't needed.

If a replica's PT is substantially earlier than RT, it will develop a large forward skew, neutralizing the problem. If a replica's PT is substantially later than RT, all other replicas will develop a skew that aligns with it. Therefore, we achieve the objective that significant skews in either direction don't adversely affect operation of those replicas or of others.

Still, a replica with a large `PT-RT` will create a large skew value for the whole group, with the legal but undesirable effect that `PT + s` differs significantly from RT. When that happens, it's important that skews don't continue to creep up, with each replica edging the others forward. A non-zero value of d helps; in our implementation we add another 500ms to the effective value of d to ensure this effect.

SIMULATIONS

The following simulations were generated from our Golang implementation.

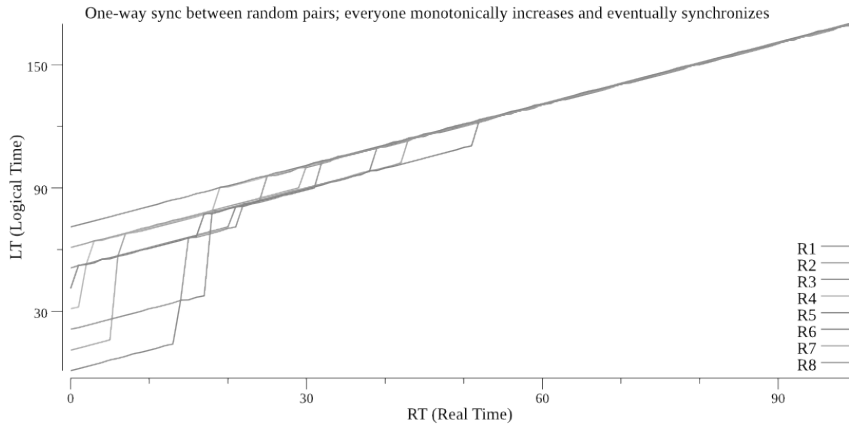


Figure 6

Convergent LT, with staggered PT

With replicas starting with PT staggered every 10-seconds, one-way-synchronizing a random pair once per second, they monotonically increase and eventually converge on the one with the latest LT (Figure 6).

Convergent LT, with variable-rate PT

With each replica's PT clock running at a different rate relative to RT, one-way-synchronizing a random pair once per second, all replicas keep converging close to the one with the latest LT, i.e. the fastest clock (Figure 7).

Far-Future replica joins, then leaves

A replica with a far-future date joins; all replicas converge on the new far-future LT by one-way-synchronizing a random pair once per second. The "bad" replica then leaves the collective. The remaining replicas have large skews, which should not change. In particular, they should not "creep up" in skew (Figure 8).

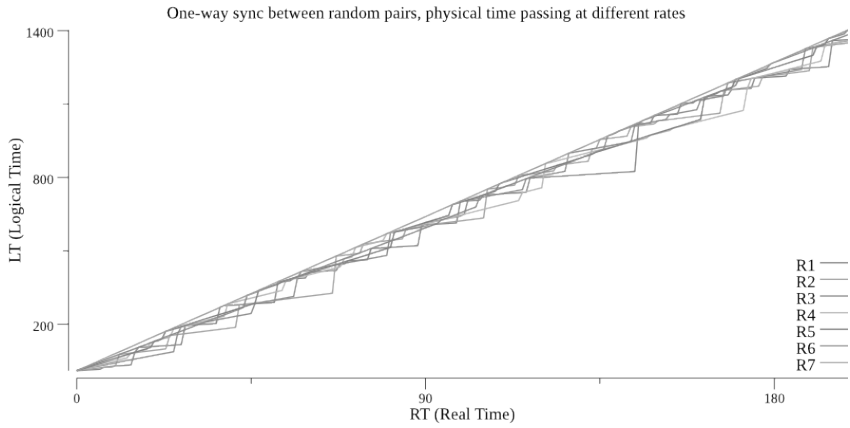


Figure 7

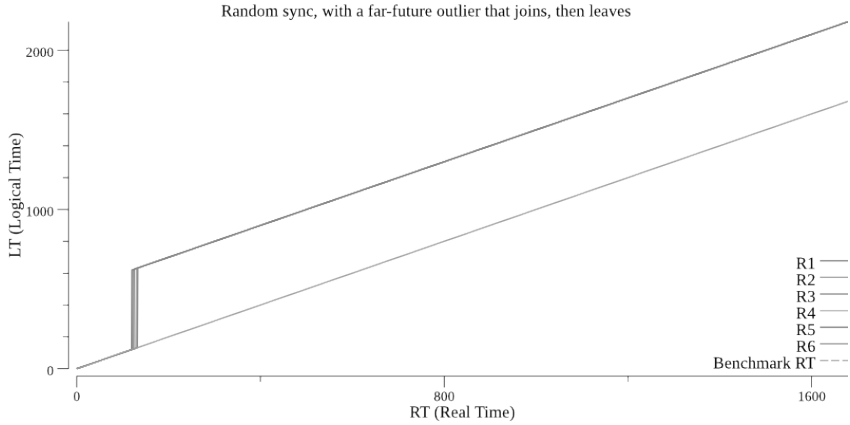


Figure 8

PROBLEMS

LT at start-up

When a replica first starts up, it will have an LT that is likely to need skew-correction. It should "fix" its LT prior to using it in a meaningful way.

One fix is to communicate with any other replica; this will bring it up to speed and set an appropriate skew.

Another fix is to persist the LT state between runs of the replica. In particular, saving the skew value. However, this is not as good as communicating with a live replica, because the behavior of PT or the collective value of the skew might have changed since the previous run.

Using LT without doing those things is still legal and consistent, but will generate events that will appear to be older than they actually are, relative to other events being generated by other replicas around the same RT.

ANTI-OBJECTIVES

The following are not goals. In some cases the algorithm gives best-effort to achieve them anyway. In some cases there are things the library user can opt-into having that goal, possibly at the expense of a different goal or constraint.

Uniqueness

The algorithm above creates locally-unique values (i.e. monotonically-increasing), but not globally-unique (i.e. two replicas can generate the same LT).

Uniqueness can be useful because it allows LT to also serve as a "name" of an event in logs or databases.

It's easy to add uniqueness. Just add more (least-significant) bits to the LT structure. Set them equal to something unique to a replica. This can be a replica ID that is unique in the world, since the rest of the Time components will never be generated again on that replica. Or it can be a sufficiently large number of random bits.

It may seem like collisions are already unlikely, however they are common under certain assumptions, namely if PT is coarsely updated and `d` is very small. Consider the example above, but rather than `d=1`, assume a fast network where `d=0.001` but a PT source that updates only once per second (on a timer, say). Once P and Q share LT, they will be identical at the same point in RT, and stay synchronized thanks to `s`. So they will likely collide if both generate an LT inside the same RT second.

Keep PT close to RT

It's nice if PT stays close to RT, but it is not a requirement.

You can achieve this, in fact making the difference bounded, if you disable skew. This keeps PT close to RT but results in a potentially large `e`, and thus you get mis-ordered events. If you accept this trade-off, you can ensure PT never strays too far from RT, as proved in the the HLC paper.[1170]

To achieve this while not giving up the objectives on the small size of `e`, you can use NTP or a similar service to keep PT close to RT.

In the end, the algorithm is simple, and perhaps even obvious in retrospect. The best things are. Simplicity is a core requirement for scalability and truly bug-free code. We hope these properties result in people using this technique.

Chapter 111:
Change: Damned if you do, damned more if you don't

Andrew Annett
@akannett

Top workplace complaints:
1) The way things are
2) Change

12:47 PM · Jun 1, 2015

credit[171]

This plays out in many ways:

1. Customers demand an improved UX,[1172] but they don't want to learn a new UX, and they always complain when it changes.[1173]
2. Team members want consistency but don't want policies.
3. Developers want to be more efficient but don't want to change how they work.
4. Strategy (p. 489) is ineffective if it's constantly in flux, but a strategy that remains unchanged in the presence of new information is incorrect.

The right choice is almost always "change." This is because change is a reaction to uncovering facts, getting smarter, or a shift in the external environment. Death awaits any organization that chooses the comfort of the familiar over the discomfort of change.

Yet, though inevitable, change is uncomfortable and exhausting. Even we who relish change, who love bragging that "it's hard but every day is different," reach a breaking point after years of adaptation and fake-gleefully exclaiming that "failure (p. 1261) is how you learn!"

Yeah, but all this learning is fricking *tiring*.

This is important for leaders to understand, if indeed "change is the only constant" as the insipid cliché goes. Even your most stoic, change-loving mortals sometimes need a break from change. Yes "it's a marathon" but sometimes you need to walk a mile to catch your breath. Look for signs of burnout or decision-fatigue, and address it proactively.[1174]

This is equally important for everyone in a startup, whether you manage others or not. Constant change can feel like management has no plan and no strategy. It takes careful consideration to distinguish between being rudderless and a culture of self-reflection and improvement.

This is exacerbated by the fact that not all change is in fact for the best. Sometimes our attempt to solve a problem makes it worse. Sometimes when we try to make code faster, we break it. The difference

is that we can see slow code objectively in the profiler and couch the code in unit tests and code reviews, and iterate before we ship, and revert to the previous version in the worst case; it's not so easy when the change is happening to a whole team, or a major product release, or a cross-departmental strategic initiative.

In fact, sometimes it's objectively impossible to know ahead of time (p. 159), and you have no choice but to place a bet.

Even deciding *what* to change is hard. Successful companies can stall out because they lose sight of the fundamental reasons they earned success in the first place—the key insights and UX of the product, or the key culture and values that attracted their first hundred or thousand employees. But successful companies also stall out because they're so dogmatic about their strategy or "non-consensus but correct" ideas that when the world changes around them, or scale breaks their previously-correct notions (p. 1345), they fail to adapt. It is not generally true that "what got us here will get us there," and that means deep change is required.

There's a mindset that everyone can use to address all of these difficulties:

Be kind.

Maybe don't judge too harshly if your organization tries to improve one thing and ends up breaking another, or where the organization takes too long to implement change. Maybe don't judge too harshly if the person to your left needs to work on something easy for a few sprints or take a vacation.

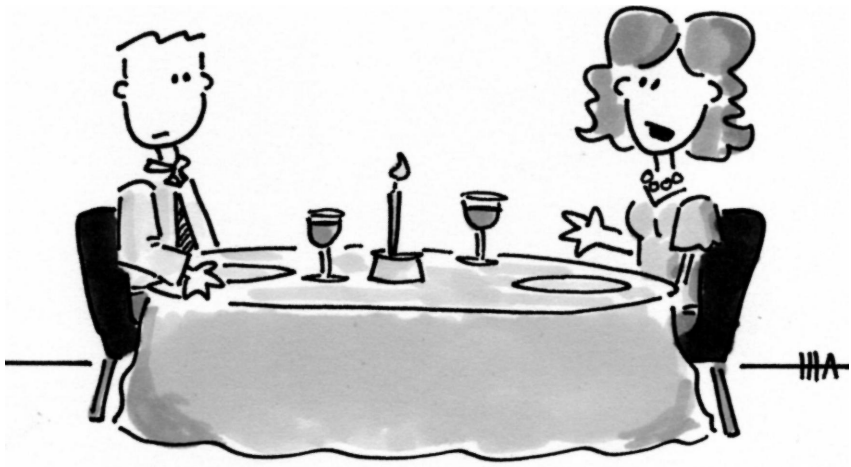
Edison had to try thousands of materials before finding the one that make lightbulbs practical. Would you have judged him for "thrashing?" Invention is often frustrating (p. 1261).

You *should* judge harshly if nobody is thinking about this. If nobody cares whether there's change or not, if there's no rhyme or reason to the company strategy (p. 489), if everyone is expected to act and feel happy and productive all of the time, then you should definitely judge.[1175] An organization that isn't striving to improve, will rot and disintegrate.

There are no straight paths in life or startups. All we can do is keep being introspective, and keep attempting the right sort of change.

Chapter 112:
How to simplify complex decisions by cleaving the facts

UPSIDE/DOWNSIDE · VETO · INVEST
FEWER DIMENSIONS · MOVE ON



"It's not that I don't like you, Ted, you just don't fit my target demographic."

credit 1176

We face complex business decisions for which there is no single correct answer, yet we must make a strong and occasionally permanent decision.

Sometimes the puzzle is created by inherent uncertainty: how a market will evolve, what competitors will do, whether new marketing campaigns will be successful, or how an important new hire will perform. There are specific strategies for dealing with inherent uncertainty (p. 193).

But I've seen little startups, mid-sized companies like WP Engine, and large companies struggle with complex decisions even in conditions of relative certainty. If this sounds painfully familiar, the following technique might help.

SEPARATE UPSIDE FROM DOWNSIDE, AND CLARIFY THE OUTCOME

The first thing is to separate the question of "how could things go right" (the upside) from the question of "how could things go wrong" (the downside). The symptom of not following this advice is arguing in logical circles. For example, take the decision "should we build feature X:"

1. If we add feature X, we will be unique in the market.
2. But we're behind in feature Y, which three competitors already have.
3. But if we build feature Y, those same competitors will then have built feature Z, and we'll still be behind.
4. But we're losing business today because we don't have feature Y.

5. But if we were unique in a different dimension, that would turn the conversation into feature X versus Y.
6. But if a competitor who already has Y ends up copying us on X, then we're back to being behind.
7. ...

All of those statements are true! That's why you're not converging on a decision.

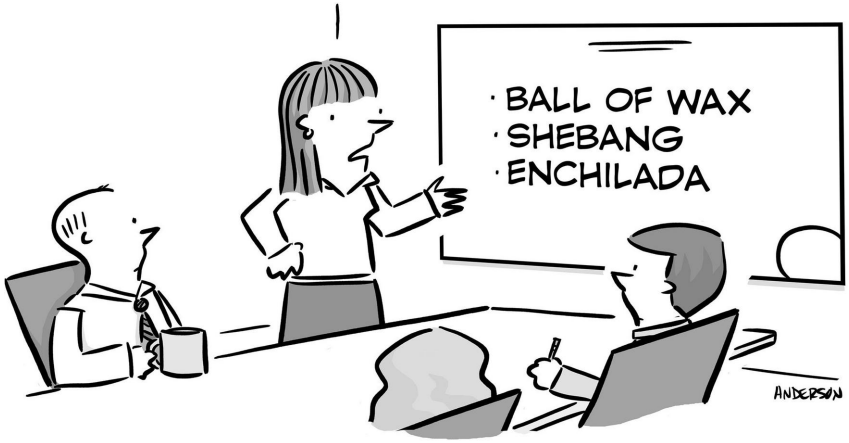
Instead, separate the upside and downside of "building feature X," and clarify the outcome of those two aspects:

Upside
Unique position in the market → deals won through differentiation.

Downside
Won't build feature Y → deals lost to a specific competitor.

When you look at it this way, the decision is clearer. If we do feature X, we earn more deals (through differentiation). Also we gain some (less measurable) brand benefit from that differentiation. If we do feature Y, we also earn more deals, but only in a subset of the sales calls, and there's no additional brand benefits. So we should probably do feature X.

A counter-argument could be: We're losing $10m/yr not having feature Y; we anticipate gaining $1m/yr if we have feature X. In this case, *quantification* of the upside/downside would also lead us to a clear winner of doing feature Y. Nevertheless, this example demonstrates that, even in the absence of data (which is frequently the case, especially with early-stage startups), this technique helps us arrive at a clear and sensible decision.



credit[1177]

"So, which 'whole' do we want to go with?"

DECIDE WITH UPSIDE, VETO WITH WEAKNESS

Let's overlay a deeper insight onto our upside/downside process. We'll also switch to a new working example, to demonstrate the universal applicability of this framework.

Suppose we're hiring a high-level, experienced position, like "VP Marketing". The "perfect candidate" is mythical—a person who is world-class at vision, strategy, operations, people, org-structure, communication within the company, communication outside the company, gets things done quickly, mentorship, etc.. So, again it's easy to go in circles when deciding on a given candidate, encouraged by her merits but worried about her weaknesses, similar to the "feature" discussion.

With leaders, however, it's you should hire based on their exceptional strengths, and then fill in weaknesses using the composition of the rest of the team. Only through exceptional strength will the person make an impact on the business; likely there's one or two critical im-

pacts you need over the next year, which you expect this person to make. You must derisk that. Here's a specific example of how to do that with Product Management (p. 817). Remember, it's your job to fill the company with people much better than you (p. 981) at every position.

So first you separate the upside ("in what aspects is she truly exceptional") from the downside ("in what aspects is she lacking?"). But now we apply a more sophisticated idea: That we must **primarily decide based on the upside** and then ask ourselves if we can **mitigate the downside**.

Specifically:

1. For the most critical business problems we have today, that this leader will be expected to solve, is she world-class in solving those particular problems?
   (Because, if not, you'll still have your critical business problems, so that's a no-hire regardless of how much you like the upsides or don't care about the downsides.)
2. For the weaknesses of this leader, do I know what those are (because if not, you can't plan for them), and are they either unimpactful, or do I understand exactly how I or they will mitigate them?
   (Since no one is perfect, there will always be items in this category. If any are impossible for you to ameliorate, it's a no-hire. Otherwise, it's the plan.)

By basing the primary decision on the upside, and using the downside only as a "veto" when it is unworkable, you have further clarified how to make the decision.

INVEST IN UPSIDE, IGNORE OR DAMPEN DOWNSIDE

Even further, we don't want to merely "base the primary decision" on the upside—we want to intentionally over-invest in it.

To explain this, and to further emphasize the broad generality in which these rules apply, let's switch examples yet again. Now the decision is: "What aspects of our product should we invest in (p. 867), over the next 12 months?" (As an exercise, afterwards try re-running this example against your own personal strengths/weaknesses (p. 569) and see if you get some insights!)

Suppose we plot our strengths in six key areas, using some relative measure (Figure 1).

People naturally focus on weaknesses. This looks like: "We're getting killed in the market for not having F" or "I'm sick of customers complaining—rightly!—that we're bad at E" or "20% of customers leave us because we're so bad at D."

As a result, we invest time and money into mitigating weakness. Sure, you can't turn a catastrophic weakness into a super-strength, but you could at least bring it to a passable neutral (Figure 2).

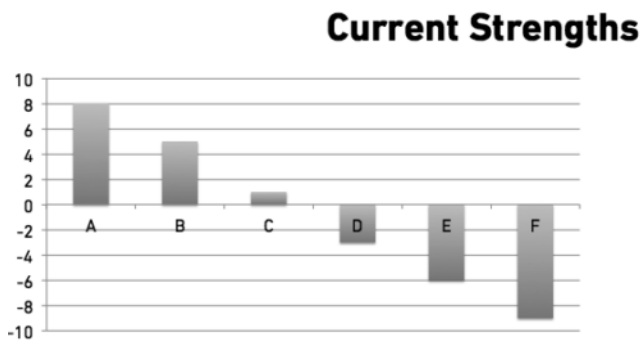


Figure 1

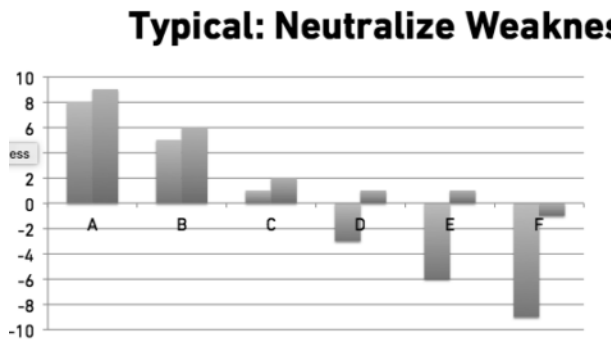


Figure 2

The trouble with this line of thinking is that it ignores a deeper truth, which is that developing a 10x strength is far more valuable than shifting a weakness to neutral. This is true personally, professionally, and in Product.

For example, in the v1 launch of the iPhone, it was far more important for it to be 10x on its strengths (e.g. form-factor and browser/email experience) than for it to shore up weaknesses (e.g. lack of copy/paste, poor call experience).

Or for example, as an engineer you will be far more effective in producing quality code quickly if you're extremely deep in one language/framework/problem-domain, than if you spread out your time becoming passable at ten different languages.

Or for example, Heroku won the hearts and minds of Rails developers because it was 10x better at the deploy/stage/production system, and therefore developers (begrudgingly!) put up with (what were *then* considered weaknesses) a read-only file system, having to use Bundler, having to use PostgreSQL, over-paying for CPU, and tying your fate to a platform.[1178]

Or as a counter-example, when we had poor web design at WP Engine (weakness), we invested and got zero return (p. 853), because it

Figure 3

turned out that our words and product/market fit were 100x more important than design.

Therefore, this is how you should be investing (Figure 3).

This drives home the idea of separating upsides from downsides, strengths from weaknesses, then further investing in your greatest upsides and strengths, while using downside as constraint to design around, or possibly a veto in extreme cases.

DECIDE USING ≤ 3 KEY DIMENSIONS

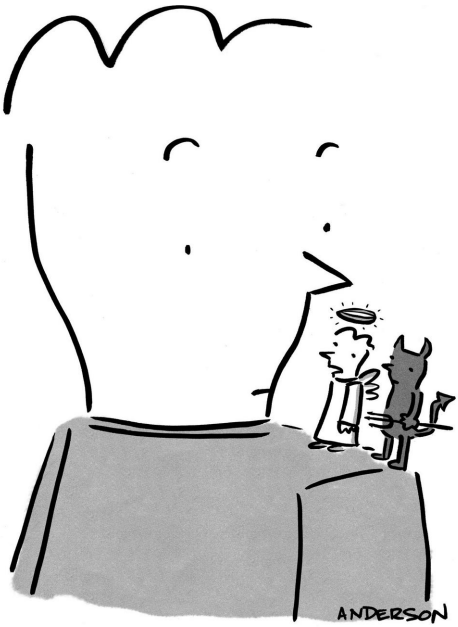
I pulled a trick earlier. It was necessary to arrive at a clear decision. You might not have noticed at the time, but you'll want to do it intentionally.

In the first example of whether to implement feature X or Y, I boiled it down to "both increase sales, but only one also increases differentiation." Of course when things are simple, the choice is clear.[1179]

But the choice wasn't so clear from the text that preceded the summary. And surely there are other considerations such as: how long will it take to implement, do we have the right team assembled, and what is the likelihood of abject failure. Why was it valid to boil it down to such a simple statement, and thus an obvious decision?

On its face, this *can't be* valid. Any simplification that ignores a dozen important dimensions can't be an accurate model (p. 465) of the problem! However, if you don't intentionally over-simplify, you will never reach a firm and clear decision.

To see why, consider the decision faced by American voters in this 2016 presidential election. Here you have two candidates which empirically are the least-liked in history. If you attempt to vote on the issues, you eventually realize there are too many to consider: climate, energy, health, taxes, economy, trade, war, education, technology, cor-



"Do it, don't do it. We don't care anymore."

credit[1180]

porations, Wall Street, abortion, drug-legalization, civil rights, and many more. It's almost impossible to agree with any one candidate on *all* the issues. And it's impossible to predict which handful of issues will actually get attention and change over the next four years; governments mostly produce gridlock, not change. This is not unlike the feature X / Y dilemma—too many considerations, and unknown which of those considerations might actually matter in the end.

What a lot of voters do, perhaps unconsciously, is select a few issues for which they posses a special affinity, and vote on only those. So for example, someone might select "elections are corrupted by money" as a key issue, and therefore support Bernie Sanders even if they agree quietly that his economic plan doesn't add up. Or someone might select "social justice" as a key issue, and therefore support Hillary Clinton (and the justices she will appoint), even if they agree it can't be appropriate to delete emails that are under subpoena. Or someone might want to "throw a bomb into the institution of government" and elect Donald Trump, even if they agree he has and will continue to say and do atrocious things.

This sort of simplification is logical, and necessary. We've talked about this before in the context of SaaS metrics (p. 833) and as a component of great strategy (p. 489). We sometimes have to be so simple as to be reductive, so we can make a clear decision.

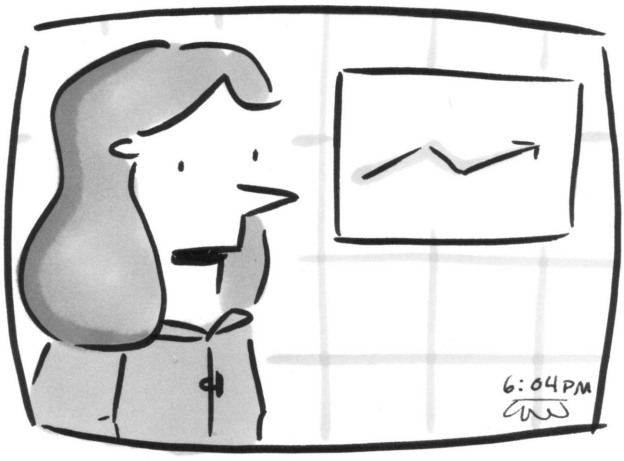
Here are two tools that use this insight to arrive at good decisions:

- Binstack (p. 603): How to make multi-dimensional choices.
- Fermi ROI (p. 171): The better way to run rubrics or "ROI" calculations.

If after the examples in those articles you're still uncomfortable with ignoring a dozen important things, consider this: If you treat all those things as "important," you'll end up with a set of "good" choices. *Any* one of them is, objectively, good. Therefore, if you further refine your process to narrow down to just one of those, *almost any process*

whatsoever is acceptable, because you're picking from a set of already-good choices!

Inevitably, however, with whatever options you excluded, you're likely to field complaints, disappointment, and argument from others who have some vested interest in the losing options, even if their interest is simply that it was their idea, or they had gotten attached to it. For them, it will be easy to whine "but how is your choice *really* better?" And they are right, because you had multiple, equally-good choices. Whatever you do, someone is going to say that. So, you should not include this "social pressure" in your decision.



"Reaction to the news was mixed, largely because we asked more than one person."

credit[1181]

DECIDE HOW TO DECIDE, THEN GET ON WITH IT

Making decisions quickly is valuable. You can usually make a different one if the first one proved wrong. (If you can't, that's a reason to take more time evaluating the decision (p. 745).)

So pick a process, use it, and move on.

Separate the upside from the downside. Clarify the choice using one, two, or at most three dimensions. Base the primary decision on the magnitude and likelihood of the upside, and use the downside to veto untenable options. Use Binstack or Fermi ROI as frameworks that guide a group to a joint, explainable decision. Remember that defining the conditions of the decision is 80% of the decision, so invest your time in that.

And don't be too hard on yourself after the fact, even if it goes badly. Life is an experiment with little predictive power (p. 429), no control group, with N=1, and which cannot be re-run.

They can't all be zingers.

Chapter 113:
What if there isn't another 10x?

"I had the whale dream again."

credit 1182

What if Dropbox is just an awesome file-sharing and backup service but not a universal key/value storage API and native document format?

What if OpenTable is the perfect reservation system but nothing else?

What if Yelp has ratings but can't solve the logistics of food delivery?

What if Facebook can make $20b/yr in advertising, but since the teens and social media elite have left, it will never make $200b/yr?

What if Twitter will never 10x its user-base but remains a critical world-wide communication, sharing, and news system for the next decade?*

In the tech industry we're constantly told that "growth" the prime determinant of the value of a business. Valuations are most often described as a multiple of revenue, with the notion that higher growth rates beget higher multiples. If your growth dips below the level of "rivets rattling out of the plane," revenue multiples plummet, and you're written off by the press as a has-been ripe for disruption by startups 1/1000th your size but "growing exponentially (p. 115)."

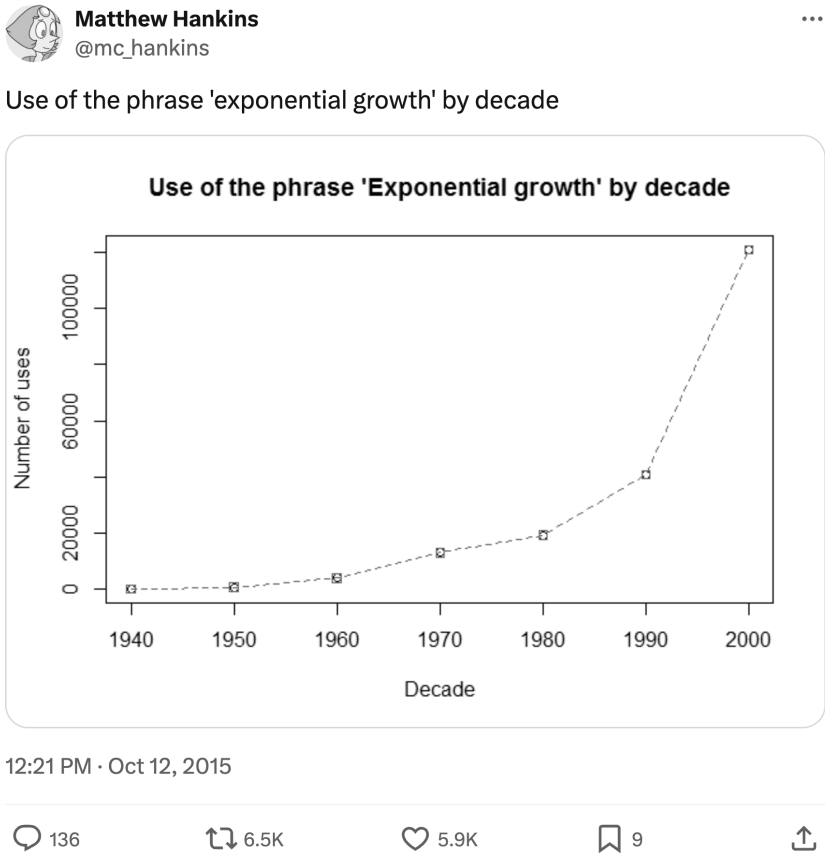
Nevermind that Microsoft created more cash profit in 2015 than Facebook made in revenue. It's only worth 12% more (in Enterprise Value), because maybe Facebook can keep tossing ads at people or invent their own virtual currency, or monetize chat users.**

"How do we 10x" is a great question (p. 839) when a startup is young. Does it always remain the right question?

---

* *Editor's Note from 2024:* Written in 2016, we can check my predictions. Indeed, the Dropbox prediction came true, as it continued to be a popular file-sharing and backup service, but none of its other products ever took off. The OpenTable and Yelp predictions proved true. Meta projects $160B for 2024, with Facebook alone being $100B of that. So this was half-right: 5x over 8 years, not 10x, but impressive. The Twitter prediction proved true, even before the Elon takeover.

** All of which they tried, and all of which failed. Their acquisitions of Instagram and WhatsApp were critically important for growth. As of this update in 2024, their efforts in the Metaverse and AI have burned tens of billions of dollars but no substantial results. Which demonstrates yet again just how difficult it is to "10x forever."



credit [1183]

At some point it cannot be the right question. No company 10x's forever. Not even Google. They know this, which is why they're now a Berkshire Hathaway-style conglomerate of individual companies which themselves could potentially 10x.

You can't know right now whether you're done 10x'ing (p. 159), but if indeed that's not a plausible result, how could you be redirecting your attention to other important things, like creating more jobs and better working conditions, to be a catalyst for employee's personal growth, to give back to relevant communities, to use your logistics

and technology and infrastructure and cost-savings to do something important with this success you've earned?

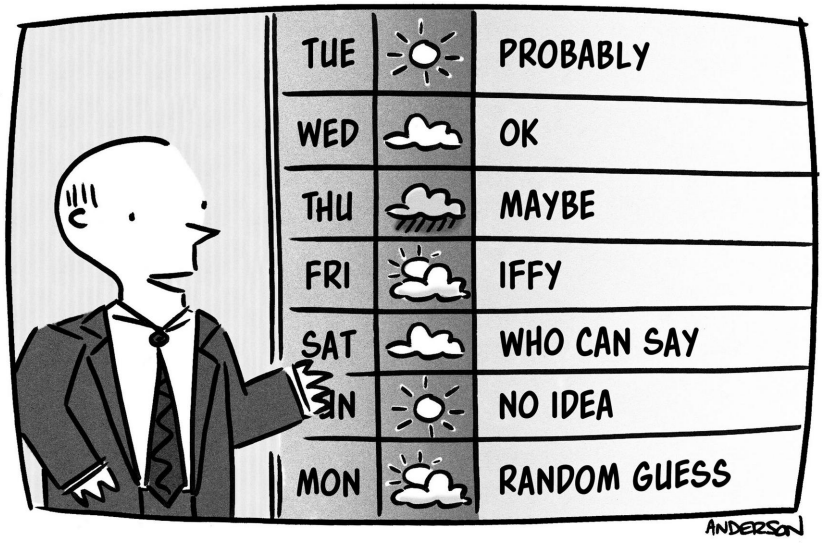
At some point, "just ploughing back into a 10x growth objective" is not the optimal strategy for making your so-called "dent in the universe."

In fact, it is at that moment, when a company is applied to betterment instead of (only) biggerment, that the nature of a company's true dent in the universe (p. 827) is determined.

What will be your legacy (p. 561)?

Chapter 114:
How to measure the accuracy of forecasts

ACCURACY ERROR · DISCERNMENT
DISCERNMENT ERROR · FORECASTING ERROR
FURTHER READING



"And now the 7-day forecast..."

"There's a 30% chance of rain today."

And then it didn't rain. So, was the forecast accurate?

Or it did rain. Is the forecast inaccurate?

How do you hold forecasters accountable, when the forecast is itself a probability? The answer appears tricky, but ends up being simple enough to answer with Google Spreadsheets.

It's a journey worth taking because of the value of building better forecasts:

- **Lead scoring**: Putting a value on a new sales lead, predicting the chance of converting to sale, and its dollar value.
- **Predicting churn**: If you could predict the chance that a given customer will churn in the next thirty days, you could be proactive and perhaps avert the loss; do this enough and you're on the road to Product/Market Fit (p. 9).
- **Predicting upgrades**: If you could predict the chance that a given customer is amenable to an upgrade, you could focus your internal messaging efforts accordingly.
- **Risk assessments**: Establishing better probabilities on risks (p. 997) results in more intelligent investments (p. 867).

So how do you measure the accuracy of a prediction that is expressed as a probability? Let's return to the meteorologist.

ACCURACY ERROR

Clearly, a single data point tells you nothing.

Rather, the correct interpretation of "30% chance of rain" is the following: Gather all the days in which the meteorologist predicted 30%. If the meteorologist is accurate, it will have rained 30% of those

times. Similarly, the forecaster will sometimes predict 0% or 10% or 50%. So we should "bucket" the data for each of these predictions, and see what actually happened in each bucket.

What is the right math to determine "how correct" the forecaster is? As is often the answer in statistics, we can take the squared difference* between the forecast and the actual result.

Suppose we have two forecasters, and we ask: Who is most accurate? "Accuracy Error" is measured by the sum of the squared differences between the forecast and reality. Whoever has the least total error is the better forecaster.

For example, suppose on some given set of days, forecaster A always predicted a 32% chance of rain, and B always predicted 25%, and suppose in reality it rained on 30% of those days. Then the errors are:

	Predict	**Actual**	**Squared Diff = Error**
A	32%	30%	$(0.32 - 0.30)^2 = 0.0004$
B	25%	30%	$(0.25 - 0.30)^2 = 0.0025$

It feels like we're finished, but alas no. If all we compute is Accuracy Error, we fall for a trap in which we easily forecast with perfect accuracy, while also being utterly useless.

* Why do we square the errors instead of using something simpler like the absolute value of the difference? There are two answers. One is that squaring the differences intentionally exaggerates items which are *very* different from each other. The other is that the mathematics of squared differences is much more tractable than using absolute value. Specifically, you can expand and factor squared differences, and you can use differential calculus. Computing a linear regression line with least-squares, for example, is derived by using calculus to minimize the squared differences, but that same method cannot be applied to linear differences. Nassim Taleb is famously against this practice, but let's not argue the point now.

DISCERNMENT

Suppose these meteorologists are in a region that rains 110 days out of every 365. That is, the overall climactic average probability of rain is 30%. A meteorologist would know that. So, a meteorologist could simply predict "30% chance of rain" every single day, no matter what. Easy job!

Our Accuracy Error metric will report that this forecaster is perfect—exactly zero over a whole year of predictions. Because, the prediction is always 30%, and indeed on those days it rained 30% of the time: $(0.30 - 0.30)^2 = 0$. Except the forecaster isn't perfect; she's not forecasting at all! She's just regurgitating the climactic average.

And so we see that, although "accuracy error" does measure something important, there's another concept we need to measure: The idea that the forecaster is being *discerning*. That is, that the forecaster is proactively *segmenting* the days, taking a strong stance about which

credit[1185]

"But, to be fair, there's a fifty-percent chance of just about anything."

days will rain and which will not. Staking a claim that isn't just copying the overall average.

There is a natural tension between accuracy and discernment which becomes apparent when you consider the following scenario:

Suppose forecaster A always predicts the climactic average; thus A has 0 accuracy error but also 0 discernment, and is therefore useless. Now consider forecaster B, who often predicts the climactic average, but now and then will predict 0% or 100% of rain, when he's very sure. And suppose that when he predicts 0% the actual average is 10%, and when he predicts 100% the actual average is 90%. i.e. when "B is very sure," B is usually correct.

B will have a worse accuracy error score, but should have a better discernment score. Furthermore, you would prefer to listen to forecaster B, *even though he has more error than A.* So the idea of "discernment" isn't just a curiosity, it's a fundamental component of how "good" a forecaster is.

How do you compute this "discernment?" We once again use squared differences, but this time we are comparing the difference between *observed results* and the *climactic average*, i.e. how much our prediction buckets differ from the climactic average, and thus how much we're "saying something specific."

For both accuracy error and discernment we use the weighted average for each prediction bucket. Let's start by computing accuracy error for forecaster B, assuming that out of 100 guesses, 8 times he predicted 0%, 12 times he predicted 100%, and the remaining 80 times he predicted the climactic average of 30%:

Prediction Bucket	Actual in this bucket	Accuracy Error
0%	10%	$(0.00 - 0.10)^2 = 0.01$
30%	30%	$(0.30 - 0.30)^2 = 0.00$
100%	90%	$(1.00 - 0.90)^2 = 0.01$

Now we weight the errors by the number of predictions in each bucket, yielding the final Error as the weighted average:

Prediction Bucket	Accuracy Error	# in Bucket	Weighted Error
0%	0.01	8	$0.01 \times 8 = 0.08$
30%	0.00	80	$0.00 \times 80 = 0.00$
100%	0.01	12	$0.01 \times 12 = 0.12$
Total		100	0.20
Average			$0.20/100 = 0.002$

Forecaster B's accuracy error of 0.002 is still low of course. Now we'll compute this new discernment score, which is exactly like accuracy error, except instead of the squared difference between "predicted" and "actual," it's the squared difference between "climactic average" and "actual":

Prediction Bucket	Actual in this bucket	Discernment
0%	10%	$(0.30 - 0.10)^2 = 0.04$
30%	30%	$(0.30 - 0.30)^2 = 0.00$
100%	90%	$(0.30 - 0.90)^2 = 0.36$

You can see that the forecaster gets no discernment "points" for predicting the climatic average, some points for predicting 0%, and a lot of points for predicting 100%. The latter is more of a departure from average, so that's really "sticking your neck out," so you get a high score for that.

Then we create the weighted average for discernment, just as we did with accuracy error:

Prediction Bucket	Discern.	# in Bucket	Weighted Discernment
0%	0.04	8	$0.04 \times 8 = 0.32$
30%	0.00	80	$0.00 \times 80 = 0.00$
100%	0.36	12	$0.36 \times 12 = 4.32$
Total		100	4.64
Average			$4.64/100$ 0.0464

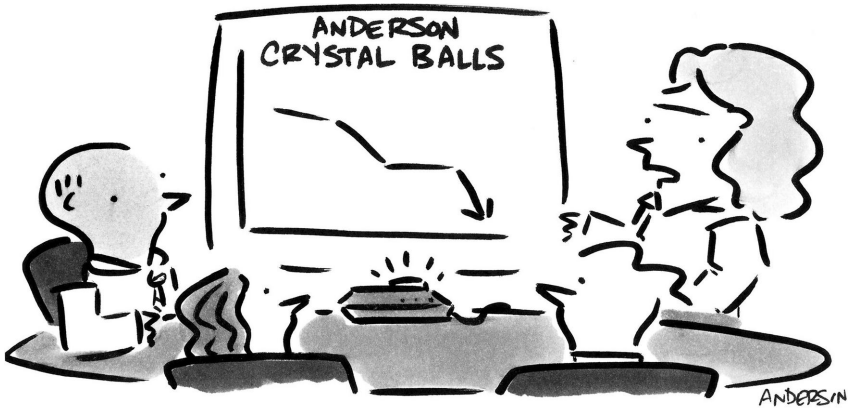
We might be tempted to conclude "there's more discernment (0.0464) than accuracy error (0.002), therefore this forecaster is better." Is that a valid way to think? What is the right way to combine these two numbers for a total "goodness" score?

To answer that, it turns out there's one more concept we need.

DISCERNMENT ERROR

Consider the life of a forecaster in Antofagasta, Chile, where on average it rains only five days a year[1187] (for a grand total of 1.7 *millimeters* of annual rainfall!). At first glance it seems easy to be a forecaster—just predict "no rain" every day.

Although that forecaster would have low error, she would also be undiscerning. But wait... how could a forecaster *ever* be discerning in Antofagasta? To be discerning you need to make varied predictions. But *reality isn't varied*, so any predictions that *were* varied, would necessarily be *wrong*! In a sense, there's no "space" for discernment, be-



credit[1186]

"Seriously?! No one saw this coming?!"

cause there's no variation to discern between. It's not fair to ask the forecaster to be more discerning than the variation that is actually available to discern.

Compare that with forecaster in Portland, Oregon, USA where it rains 164 days out of the year—about 45%. And there's no "rainy season"—it's just chaotic. Here even just predicting 55% or 35% here and there could still be highly accurate but increase discernment. And a world-class forecaster has the space to create a significant amount of discernment.

So it's not quite fair to ask "How discerning is the forecaster?" Instead we should ask "How discerning is the forecaster, *compared with* how much discernment is even *possible*?"

The maximum amount of discernment possible, given the climactic average c, is:*

* For the mathematically curious: Maximum discernment happens when the forecast is only 100% or 0%, and when it is completely accurate. If there were N predictions, the positive case will happen Nc times, and the negative case $N(1-c)$ times. Discernment in the positive bucket is $(1-c)^2$ and in the negative bucket $(0-c)^2 = c^2$. The weighted average is $\frac{1}{N}[Nc(1-c)^2 + N(1-c)c^2]$.

$$c(1-c)$$

In general, the closer the climactic average is to 0% or 100%, the less discernment there can be. The maximum possible discernment of 0.25 is available when the climactic average is 50%, i.e. it's a coin flip.

In the 30% example, the maximum possible discernment is $0.30(1-0.30) = 0.21$. Forecaster B's discernment of 0.04 is therefore not too impressive—plenty of room for improvement. Although still of course B is better than A, who had no discernment whatsoever.

In the case of the desert, with a climactic average of $5/365 = 0.013\%$, there's only 0.0128 potential discernment available.

In any case, this allows us to compute a metric that is comparable to error, but for discernment:

[Discernment Error] = [Maximum Discernment] – [Discernment]

That is, if you have no discernment, that's another type of "error"—your forecast is lacking the descriptive power that would come from being maximally discerning. The more discernment you demonstrate, the less of this "discernment error" you exhibit, and the better your forecast is. Just like the less accuracy error you have, the better your forecast is.

PUTTING IT ALL TOGETHER: THE FORECASTING ERROR SCORE

It turns out** you can simply add accuracy error to discernment error, and arrive at a rigorous metric:

Factoring out the common $Nc(1-c)$ and canceling with $\frac{1}{N}$, we're left with $c(1-c)[(1-c)+c] = c(1-c)$.

[Forecast Error] = [Error] + [Discernment Error]

Or, writing out each component:

[Forecast Error] = [Error] + [Maximum Discernment] – [Discernment]

Here's a way to see why this math works: Every forecaster's baseline is to guess the climactic average. That will get you a total score of $c(1-c)$, because you have no accuracy error, but maximum discernment error.

From there, forecasters will try to deviate from the climactic average. The more they put themselves on the line, the less discernment error they rack up, however they also have to be right! The best forecasters outperform the climactic-average (discernment) by more than the accuracy error they introduce. The overall score tells you who is better. Lower is better, since the score is a "total error."

It is possible to do worse than the climactic average—to make guesses, but be wrong. You know that's happening when the total error is larger than the baseline $c(1-c)$.

Indeed, an alternate and equivalent scoring method divides Forecast Error by $c(1-c)$, to create a metric where 1 means "equal to the baseline," 0 means "perfect forecast," and the amount greater or less than 1 indicates proportionally how much better or worse the forecast is from baseline. This has the virtue of the reader not needing to know the value of $c(1-c)$ in order to understand whether a forecaster is better or worse than the baseline, and by how much.

While the total score is useful, the individual components of accuracy error and discernment are also useful because they help you analyze what's going on:

** See "Further Reading" below for the mathematical justification.

Accuracy Error	Discern	Meaning	Explanation
↑	↓	**Failure**	You're not segmenting the population, and yet you're still worse than guessing the climactic average.
↓	↓	**Useless**	You're only accurate because you're just guessing the average.
↑	↑	**Try Again**	You're making strong predictions, so at least you're trying, but you're not guessing correctly.
↓	↑	**Ideal**	You're making strong predictions, and you're correct.

Now that you know how to measure forecasts, it's time for you to build some forecasting models. So go try to better understand your customers and prospects, and use this math to know whether iterations of your model are improving.

FURTHER READING

- Glenn Brier's original paper[1188] proposing this method in 1950, but without this three-component breakdown that was discovered by Allen Murphy in 1973.
- Brier score[1189] on Wikipedia: A more formal explanation including the three-component breakdown (which are labelled and explained differently from my exposition, but which are mathematically identical).
- Stein's Paradox:[1190] An estimator that's always better than the historical average, but in a way that apparently can't be true.

Chapter 115:
Darwin doesn't automatically select the best companies

credit[1191]

"It's a fad."

"Survival of the fittest" is not the same thing as "survival of the best," though this is not obvious at first glance.

"Fittest" means "most prolific at becoming grandparents." Sound weird? You might think "fittest" means "most prolific at creating offspring," but that's not true if the offspring themselves are unable to reproduce. A horse mated with a donkey creates a mule that is sterile; the line ends.

The "grandparent" requirement is the *only* requirement. There's no other notion of "good" or "bad," like humans have with ethics or how we want our companies to work. Which means "bad" things can be just as "fit" as "good" things. Viruses are amazingly fit—becoming grandparents at prodigious rates, quick to spread and difficult to stop once noticeably numerous.

That last bit should sound familiar if you follow Theories of Startup Laws & Metrics. Investors and founders alike have been obsessed about "growth at all costs," just as a virus colony grows without care to its effect on its host cells or whether the colony's growth ends up destroying the host body completely, taking the colony down with it in a viral version of Lord of the Flies. Similarly, we hold up "viral growth"[1192] as the best sort of growth, because exponential curves overwhelm all others (p. 115), thus once the mechanism is stable and the company gets large enough to be noticed, it's neigh-impossible for competitors to catch up.

There's nothing inherently wrong with obsessing over growth. In winner-take-all markets, growth is objectively the correct goal.

The problem is "growth *at all costs*." **When "growth rate" becomes the *only* important metric for company "fitness," other metrics are left unsolved.**

"Profitability" is perhaps the most-talked-about example of a metric left unsolved. Many companies who won the growth battle never figure out how to convert themselves from a virus to a business. Some already passed into the zone of "can never be profitable unless something drastically changes," (e.g. GroupOn, Zynga, SoundCloud), whereas many are able to continue due to massive investment, and

the jury remains out on whether they can ever produce profits (e.g. SnapChat, Pinterest, WeWork, InstaCart, Square, FourSquare, and even Twitter, who has impressive revenue but has still never been profitable). Presumably, some will and some won't ever figure out a business model, but the point is that their position as "winners of the market" is due to being an excellent virus in a Darwinian sense, not due to possessing an excellent business model.

"Betterment of humanity" is an even less-talked-about example of a metric left unsolved. Libertarians might not care whether a company earns a profit by making the world a little bit better or extracts a toll by making it a little bit worse. But it seems to me that it's our duty as entrepreneurs to create companies which leave the world better than we found it, or at least not make it worse, and our duty as employees to invest the majority of our waking hours and the most productive years of our lives in companies that make the world better, not just "hit a number."

Many of the market-winners *do* make the world better. Despite famous setbacks, advances in biotech over the past few decades have resulted in longer and better quality of life. The success of WordPress world-wide (25% of all domains) makes it orders of magnitude cheaper to completely own your website, without which you are invisible and voiceless. Amazon's cloud computing services enable the creation of technical products and services which would be impossible to create a decade ago, both for new startups and 100-year-old companies still reinventing themselves into relevancy.

But many of the winners are a mixed bag. Uber makes so-called "entrepreneurs" out of under-paid employees who were indentured servants to the taxi medallion system, but they are still indentured and still under-paid. Twitter connects people and ideas but kills attention spans and stokes hate more than debate. Google puts the world's information at our fingertips but destroys personal privacy in ways more insidious than most people want to know. Self-driving cars will solve traffic, dramatically improve the environment, provide transportation for people who can't or shouldn't drive a car, and could even trans-

credit[1193]

"While I understand your concerns, I like to think that by creating shareholder value we are delivering a kind of justice."

form the landscape as parking lots and garages become obsolete; they will also destroy millions of jobs in the transportation industry, many of them high salaried relative to the job-holders' level of institutional education.

Darwinian growth wins by definition; there's no arguing that, no use crying about it, and government regulation will only delay the inevitable while shifting the location of the winners to other countries.

It's up to we entrepreneurs and employees both, to ask whether growth is *sufficient*. Whether we stop at growth metrics or whether we accept additional objectives. Whether our companies will uphold other values as *also* being inalienable and unassailable, allowing those additional values to drive our most important decisions, and asking growth to be compatible with those values.

The real question is not how to retard growth in submission to other values, but rather how to use our insistence values to drive growth.

Fortunately, there are many examples to inspire us:

Small
Buffer[1194] is a 5-year old company with profits and growth, in the overcrowded space of "Social Media Tools," with a product that's not particularly special or unique (sorry guys, love you but it's true!), where talent nevertheless beats their door down to work there even in cases where they know ahead of time they'll take a pay-cut (because salaries are set by a formula[1195]). This is not because Buffer's product is "changing the world" but because of the company's devotion to complete transparency and positivity, being honest about struggles and successes, and of employees' health and well-being. This is growth driven by values.

Mid-Sized
TOMS[1196] shoes is a massive success, not because of a superior product, but because of a viral story that people want to support and retell, exactly because its mission is so commendable and pure. This is growth driven by values.

Large
USAA[1197]—with the incredibly boring home page title of "insurance, banking and retirement solutions"—is the 49th most valuable company on Earth with $24 billion in annual revenue and $3 billion in profits. Their customer service consistently earns NPS between 75 and 80—some of the highest scores in the world—so their financial health is matched by customers who truly want the product. USAA is well-known for being one of the best places to work, even in traditionally low-paying, low-formal-education positions such as customer service, and for their overwhelming and unwavering commitment to families of the US Armed Forces. This is growth and profit driven by values.

In short, you can grow and be profitable exactly because of goodness and love (p. 275) (That article details many ways to do it.)

Building a virulent company is wonderful, but only if the virus carries beneficial effects—creating customers who love the service, creating employees who enjoy spending most of their waking hours at work, and creating effects to our economies, communities, and societies that are net-positive.

When you build your unstoppable company, make it an unstoppable force for good.

Chapter 116:
Refutation: An acquisition is always a failure

"I'm sorry, Ted, I think we need to slow down. I've been hurt in mergers before."

Oh how the media loves superlatives (but only because that's what we click on and share).

Jake Lodwick wrote an article[1199] on PandoDaily entitled **"An acquisition is always a failure."** He explains:

> Either the founders failed to achieve their goal, or—far likelier—they failed to dream big enough. The proper ambition for a tech entrepreneur should be to join the ranks of the great tech companies, or, at least, to create a profitable, independent company beloved by employees, customers, and shareholders.

On his startup's sale to a larger company, he then laments that the "youthful energy that created so much value was siphoned off."

These are common sentiments, but they're the false wisdom of those who haven't experienced the alternative. It's classic founder naïveté to think that "youthful energy" would have been maintained had the company remained independent.

Companies are constantly changing. Cool ones cease being cool, new products cease being new, the slow-growing startup with a terrific idea is eclipsed by the faster-growing one with weaker ideas, the fast-growing startup by definition changes itself so quickly (p. 827) that year-over-year it's unrecognizable to those inside it.

WP Engine is one of those, and we laugh about how drastic the changes are, in the rare spaces between tackling the latest challenge. We have "youthful energy" for new innovations, but we also have all the challenges of fast-growing companies with hundreds of employees; that's part of success. A company won't be a band of six people doing whatever they want forever, whether you sell or stay independent.

It is the nature of things to change; expecting otherwise is foolish.

The other fallacy of the article is that his sale was a failure because the founder didn't like being an employee at a larger company. More likely, the typical postpartum depression that naturally follows even the most successful of exits (p. 1005).

But what about the other employees? Did all of them hate life too, or did they have new opportunities they didn't have previously, did they get better compensation and benefits, did they have peace of mind with greater job stability? Most people don't have a founder's temperament for independence and risk. What if those people *wanted* those jobs? What if more jobs are created, faster, due to the investment by the acquirer?

Sure not all acquisitions go that way. Sometimes the product is ruined, sometimes the culture is annihilated, sometimes jobs are destroyed, sometimes it's an unmitigated disaster.

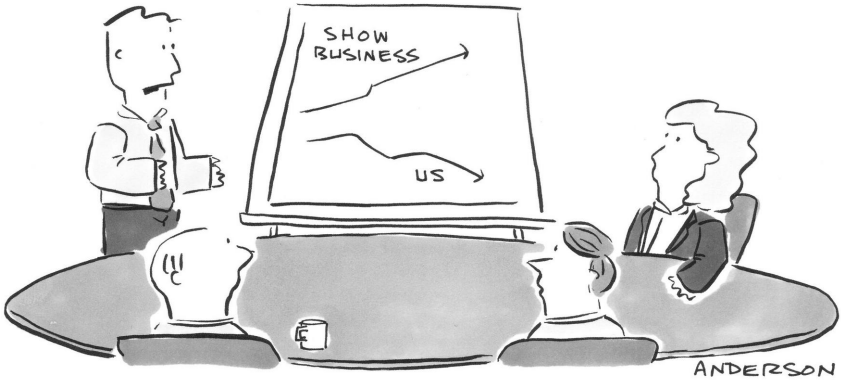
Of course, all those things happen at independent companies too. People love quoting statistics about how most acquisitions are failures, but they forget to mention that most independent startups also fail, or at least cease over the span of time that we measure acquisitions.

I know of quite a few acquisitions where nearly everyone stayed for years afterward, where it was the right thing to do in retrospect, where there were new opportunities for growth and enhancement because they were part of a larger company, and where all the employees made good money both in the acquisition event and in being well-compensated afterwards. In many of those cases the founders left ahead of others, and yes they went back to founder-y types of activity. That doesn't prove it wasn't the right choice at the time, both for them and their employees and products.

Do you think Google's acquisition of DoubleClick or YouTube was really bad for everyone there? Or eBay's acquisition of PayPal? Or the hundreds of small companies who weren't "killing it" as much as their hype and Twitter feeds suggested, but who now are on solid teams with nice stock options in successful tech companies?

How about instead of superlative language, insulting everyone who makes the choice to join forces with a larger company while providing liquidity for employees who also took a risk, we ask "When is acquisition wise?" and "What are the conditions under which acquisitions are likely to be good for all parties?"

Chapter 117:
The wrongness of relativism



"But, to be fair, there's no business like show business."

Instagram had 13 employees when they sold for a billion dollars. **Was there any startup founder who didn't feel depressed when that happened?**

Yes, *depressed*. All those years of toil and sacrifice, the pain of scaling a complex organization and a complex product, scratching and clawing for revenue growth month after month, and you'll never have that sort of exit, and 26-year-old Kevin Systrom did it in only 2 years, with 13 buddies, with $0 in revenue, never having had to worry about how to make money or gross margins or hiring at scale or making sales calls or running out of money.

"It's not fair," we all said to ourselves.

That was before WhatsApp sold for 19 billion dollars, with 55 employees. That's $345,000,000 per employee, compared to Instagram's paltry $77,000,000 per employee. So much better than those chumps at Instagram! Well, at least they had to toil for four years instead of two. Still...

Now* Slack has been valued at over a billion dollars, less than a year after launch, every month adding more than a million dollars of annual recurring revenue. They're next! Another collective sigh shudders through the ranks of startup founders, as we grumble over our trifling grindstones.

Oh, but don't worry, the pundits have wasted no time perpetuating this emotional insanity back onto the founders of these very companies! As IronFire VC Eric Jackson put it 18 months later:

> "Systrom has to be feeling like he totally missed this wave. Instagram likely worth $15B today minimum." *(This*[1201] *was the original tweet; it's been deleted.)*

You know, Kevin, *minimum*. It's your turn to be depressed, and dammit you had better *take* your *turn*!

---

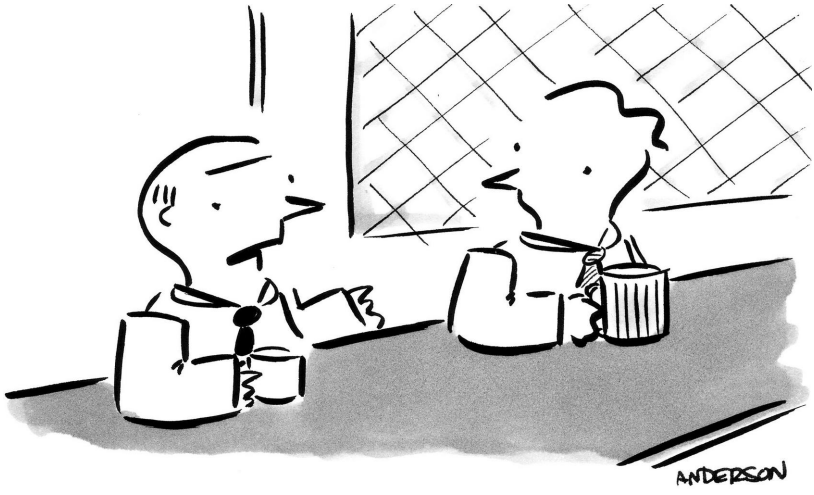
* This was written in 2014.

Of course you can see through this psychosis. You mustn't judge yourself with the yardstick of other companies, particularly not against the rarefied few who dwell in the extremities of reality.

Measuring yourself against the average or median isn't better. Who cares whether the "average" company is P whereas you're just Q? For every such condition, I can show you another company in an even worse condition with regard to that metric, but which is extremely successful because of other reasons. And each of you would envy each others' strengths while ignoring your own (p. 569), all the while getting increasingly depressed (p. 1407). Sometimes, benchmarks aren't healthy.

The right thing to ask is: "What is the best thing I can do for my company, with the specific challenges and benefits that I have in front of me (p. 1065)?"

It doesn't matter whether we're talking about valuations,[1203] growth rates (p. 1191), retention rates (p. 335), NPS scores, early-stage uncertainty (p. 429), ratio of revenue to employee, CAC (p. 1377),



"Explain to me how comparing apples and oranges is fruitless."

credit[1202]

cash-burn, LTV (p. 1355), gross margin, or selling your company (p. 1005).

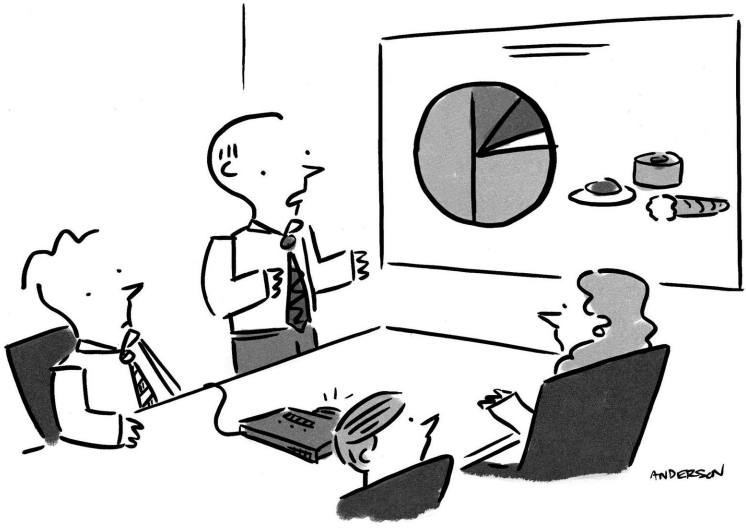
The question isn't "What is the average / extreme from some other company," but rather, "Am I happy with this metric, and if I change it at the expense of other metrics, or of innovation, or of my lifestyle, or of my happiness, is that the right trade for me (p. 861), for the company, for my customers, and for my employees?"

If you can answer in the affirmative to all that, you have your answer.

If you're not even sure how to reason in this manner, *that* is your next big challenge. Because, if you're not confident in reasoning about yourself, you have no choice but to compare your worst fears (exaggerated negative) with the public façade of other people (exaggerated positive), and to the absurd extremities that are the darlings of the news media, and this is not a path to false Impostor Syndrome (p. 457), not a path to finding good answers.

You're the captain of this ship (p. 827), and you're navigating your own waters. Don't forget it.

Chapter 118:
Rare things become common at scale



"The idea is to not only grow our piece of the pie, but to expand into other nearby pastries."

Something interesting happens when you run more than 1,000 servers, as we do at WP Engine,* powering hundreds of thousands of websites.

Suppose that on average a server experiences one fatal failure every three years. The kernel panics (the Linux equivalent of the Blue Screen of Death), or both the main and redundant power supply fails, or some other rare event that causes outage. This isn't a quality issue—this is normal. This isn't something to "fix."

> *"Windows NT crashed.*
> *I am the Blue Screen of Death.*
> *No one hears your screams."*
> *–Haiku from FSF*[1205]

But remember, we have 1,000 servers. Three years is about 1,000 days. So that means, on average, **every single day we have a fatal server error.**

Not to mention 10 minor incidents with degraded performance, or a DDoS attack somewhere in the data center affecting our network traffic, or some other thing that sets pagers a-buzzing in our DevOps team and mobilizes our Customer Support team to notify and help customers.

"Well sure," you say, "that's normal as you grow. If you had just 10 servers and 100 customers, you'd have fewer problems and many fewer employees. Today you have more customers, more servers, and more employees. What's so hard about that?"

---

* Editor's Note: As of 2024, this article is ten years old; we now run twenty times as many servers, and the lesson of this article continues to be accurate.

credit[1206]

"Boy, what a day, huh?

The insight is that **scale causes rare events to become common.** Things happen with 2000 servers that you *never saw even once* with 50 servers, and things which used to happen once in a blue moon, where a shrug and a manual reboot every six months was in fact an appropriate "process," now happen every week, or even every day.

Things as rare as, well, you know...

It's not only *problems* that morph with scale, but **your ability to handle problems**.

For example, a dozen minor and major events every day means 20-50 customers affected every day. Now consider what happens as we try to inform 50 customers. For some we won't have current email addresses, so they don't get notified. Some of *those* will notice the problem and create extra customer support load; at worst they'll post on Twitter about how their website was slow or offline today and WP Engine "didn't even know it." Then our social media team has to piece all this together, attempt to respond, maybe put together a special phone call with that customer, and so on. Those customers are also

more likely to leave a bad review on some review site, compared with the 99.99% of customers who experience no such incident, but also had no reason to decide that "today is the day I will go to a review site and leave a good review."

Or consider the scale-ramifications of on-boarding 1,000 new customers a month. In that case, it's likely that any given server issue will affect a customer who has only been with us for a month or two. Thus the issue causes a "bad first impression," which is harder to address than a customer who has been with us for three years and has built up a bank account of patience.

So, rare things being common isn't just difficult from the operational side, but also when you try to handle those problems with customers or other downstream consequences, causing *much* more work to solve than when the company was small.

The usual response to this is "automate everything."

As with most knee-jerk responses, there's truth in it, but it's not the whole story.

Sure, without automated monitoring we'd be blind, and without automated problem-solving we'd be overwhelmed. So yes, "automate everything."

But some things you can't automate. You can't "automate" a knowledgable, friendly customer support team. You can't "automate" responding to a complaint on social media. You can't "automate" the recruiting, training, rapport, culture, and downright *caring* of teams of human beings who are awake 24/7/365, with skills ranging from multi-tasking on support chat to communicating clearly and professionally over the phone to logging into servers and identifying and fixing issues as fast as (humanly?) possible.

And you can't "automate" away the rare things, even the technical ones. By their nature they're difficult to define, hence difficult to monitor, and difficult to repair without the forensic skills of a human engineer.

Does this mean all our customers have a worse experience? No, just the opposite. Any *one* customer of ours has *fewer* problems per

year now than a year ago, because we're constantly improving our processes, automation, hardware, and human service. It's when you look across the entire company, and the non-linear additional effort it takes to not just improve the average experience, but to manage the worst-case experience, that you appreciate the difficulties.

This explains the common effect where people complain about a company every day on Twitter, yet you yourself have never had an issue with them. The paradox is solved by realizing that "rare things" means *you* probably never experienced it, but at scale, *someone* is experiencing it each day.

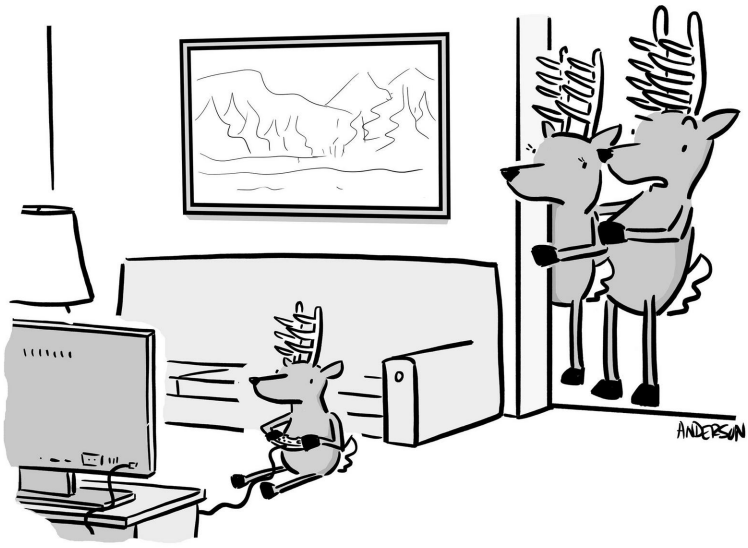
Does that give high-scale companies like WP Engine an excuse to have problems? No way! In fact, if we're not constantly improving on all fronts, the scale will catch up and overtake us.

But for those of you in the earlier stages of your companies, when you project 5x growth against 5x costs (or only 3x the costs because you'll get cost-savings at scale), you're guessing low. When you show 5x growth in projections but don't budget for new hires in areas like security, technical automation, specialized customer service areas, and managers and executives who have trod this path before and come battle-hardened with play-books on how to tackle all this, you're heading for an ugly surprise.

And with high growth, the surprise appears quickly, and recovery means acting twice as fast again to claw out of the hole and then finally get ahead of it.

Scaling is hard (p. 773)!

Chapter 119:
Ho Rudolph



"I understand playing reindeer games, but watching other reindeer play them…"

credit 1207

Ho Rudolph—fog is thick tonight.

Can you ride point?

For the children.

Why is this email https://three.sentenc.es?[1208]

Ho Santa,

Oh damn, it's going to be like that.

All those years of name-calling, ostracized, excluded, and now, with NO NOTICE whatsoever, no "Hey Rudoph, how's it going, how's that new Klondike strategy of yours coming along"—now you NEED me?

Wait, wait, let me guess—I'm supposed to say "Oh Santa, thank you *so much* for the *opportunity*, I'm so grateful for *finally* being *included* and being *useful* to Your Great Rotundity, I'm so *honored*!"

Maybe you and the other reindeer should have THOUGHT ABOUT THAT while you were calling me Pinocchio for the past 18 years. Maybe you should have THOUGHT ABOUT THAT while excluding me from every game of kick-ball and Parcheesi and Cards Against Caribou.

I'm pretty sure I wouldn't even like Parcheesi, but *how would I even know?*

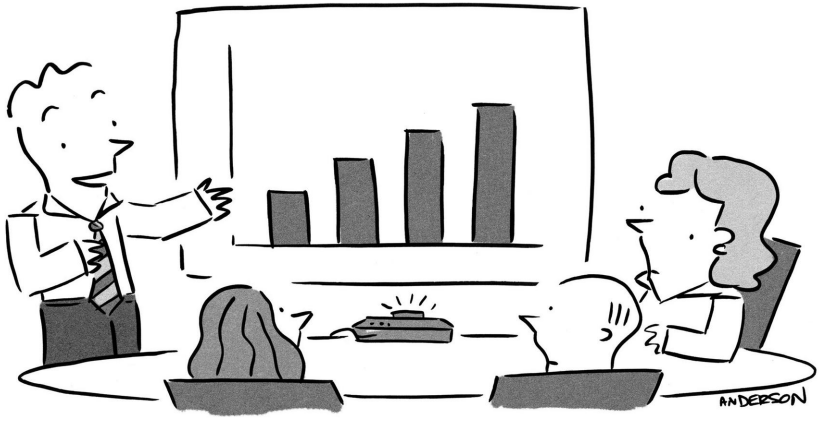
So YEAH, I think I'll PASS on your "generous offer" of "upgrading" my reputation to "living foglight." I'm sure everyone will LOVE me then, right? This is my big turn-around moment? I have better things to do with the next 24 hours than taking sleet to the face while eight of my new "chums" stare at my ass and sleigh bells shatter my eardrums.

P.S. Throwing the children in there was a low blow. That's a you problem.

Excuse typos, all hooves from my iPad.

Chapter 120:
Why I don't like the LTV metric (LifeTime Value)

LTV · NOT-LTV



"What's really exciting is if you extend this out five years, we own around 400% of the market!"

credit 1209

A metric is a single number stuffed with an enormous complexity of causes, effects, and interconnected processes. This is a two-edged sword.

This is powerful because it lets you reason about complex systems: how they look now and how they are changing. It helps you focus on what's important.

But this is dangerous because you're combining so many disparate and disjoint processes and systems that the number has little precision, and no explanatory power. And then you *think* you understand something that you don't. That's how bad decisions are made with confidence. The rubric (p. 171) always gives you 8 decimal places of "precision," even if those digits are all error (p. 913) and uncertainty (p. 193).

> *"The greatest enemy of knowledge is not ignorance. It's the illusion of knowledge."*
>
> *—Stephen Hawking*

For many SaaS businesses, the incorrectness of the LTV metric outweighs the value it supposedly confers.

The definition of LTV is the "Total gross profit the customer will generate over its lifetime." "Gross profit" means "Income after deducting expenses that are required to deliver the product," including tech support and infrastructure (for SaaS products) and cost of materials (for physical products). It does not include the costs of sales and marketing, nor the costs of development, or any other costs of running the business.

The formula is:

$$\text{LTV} = \text{MRR} \times \text{GPM} \times \text{Lifetime}_{\text{Months}}$$

Or, since typically we're estimating the average lifespan of a customer from our monthly cancellation rate:

$$\text{LTV} = \frac{\text{MRR} \times \text{GPM}}{\text{Cancellation}_{\text{Monthly}}}$$

LTV is commonly used to determine how much you can spend to acquire a new customer (CAC). In particular, if it costs more than LTV, then the business is unprofitable even at scale, and over ten years. Whereas if, say, $\text{LTV} = 3 \times \text{CAC}$, then even if we're unprofitable at first (as it takes months of revenue to pay back the cost to acquire the customer), we will be profitable in the long run (as a large percentage of our customer base has already paid back CAC). (See this article on annual pricing (p. 353) for pictures of this effect, and how different pricing can change the effect.)

Using LTV in a business decision—like how much we can spend to acquire a new customer—implies that the lifetime gross revenue from a customer is know-able today. Clearly, though, it isn't.

Every component of LTV changes over time:

- MRR—changes due to how systematic you are at upgrades, your ability to cross-sell, growing/shrinking within each customer, per-use charges, the percentage of the customer base belonging to different customer segments.
- GPM—efficiency of the service, which for small companies can change by 30% in a year and even large stable companies can move by 1% per year.
- Cancellation Rate—hopefully shrinking as the company improves (p. 335) the product and service to address the causes of cancellation, but over a time-frame of years, this can change dramatically with advent of new competitors, shrinking market, different technology, or mixing different customer demographics as you grow into adjacent markets (p. 793).

For example, Hubspot famously had a low LTV, but increased in 3x in 18 months.* That's a big swing in a metric that purports to summarize the next ten years.

Three variables, all changing, unpredictably, which you multiply together and.... you expect the result to *mean* something?

When you treat a number as being stable and solid, when in fact it can vary by 2x in a year through normal business activity, you make poor decisions. So for example when you read "An LTV:CAC ratio of 3:1 is healthy," if your LTV metric can't be trusted, neither can that formula. Your might believe you're being efficient in acquiring customers, only to find your growth is capped (p. 1153), or that you're not profitable even at scale (p. 1087).

* Documented in David Skok's famous SaaS metrics overview.[1210]

credit [211]

"So you gonna rush in, or what?"

WHAT SHOULD YOU DO INSTEAD?

Every "insight" that LTV is supposed to give you, you can get from different metrics that work better, because they don't involve long time horizons. And they point to actions you can take.

For example, take the original question: "What is a reasonable CAC?" The typical answer is "LTV/CAC should be at least 3 for healthy companies, and 5 is very good."

But you can also compute what CAC is "reasonable" by thinking in terms of pay-back period $p = \frac{\text{CAC}}{\text{MRR}}$, which is the number of months it takes before a single customer becomes profitable, because we have received (in revenue) the cost (in CAC). Or better, in my opinion, $p = \frac{\text{CAC}}{\text{MRR} \times \text{GPM}}$, so that we're accounting for the costs to serve those customers.

This payback period is also useful to see why annual prepayments are far smarter than you might think (p. 353), and for the metric COC

(Cost Of Cancellation) (p. 1361), which leads to more insights into how cancellation affects the cost of growth, and for the Max-MRR metric (p. 1153), which leads to insights into how cancellation affects your maximum size. LTV doesn't lead us to any of those insights.

A good rule of thumb with pay-back period is that 6 months is fine, 3 months is fantastic, and 12+ months is poor unless (1) there's indirect strategic benefit, e.g. branding, (2) efficiency is improving so we want to stick with it, (3) it is a mature company or with larger business customers where you can justify an assumption of 5-10 years of revenue per customer.

Another use of LTV is: A measure of dollars earned by the company over the long run, and thus something that ought to be going up over time. True, but in practice I find you *always* need to know the values of the individual components to truly know whether the company is healthy.

For example, if LTV is steady, is that OK? If all the components are steady, maybe that's OK. But what if GPM is improving due to investment in cost-cutting measures while cancellations are increasing, and thus LTV is stable. Is that good? Heck no! Your customers are pissed. Or what if MRR is increasing because you're landing larger customers, but GPM is decreasing because you're very inefficient at serving them? That's both good news and bad news, and points out that you need more efficiency work right now, or you need to raise money so that you can remain devoted to growth for the next few years.

Thus, measuring MRR, cancellations, GPM, and CAC individually are always necessary. Sure you can combine them into a number, but I think that only serves to hide data, hide insights, not help "get a handle on the business."

Chapter 121:
COC: A new metric for measuring cancellations in SaaS business models

COST OF A DOLLAR · PAYBACK · COC · HIGH COST
MAJOR EXPENSE · NET CHURN · NEW STANDARD

"I used to have a lot of issues, but then I just cancelled my subscription."

credit 1212

You probably know that high cancellations kill growth (p. 348) in scaling SaaS companies. You also probably know why: Acquiring new customers scales with Marketing and Sales, which is linear, whereas cancellation rate scales *exponentially* with your size (hence phrases like "3% per month"). Cancellations win.

I have a better, clearer way to both visualize and measure this financial effect, specifically the *cost* of cancellations in the language of *acquiring new customers*. This puts a tangible dollar figure on cancellation, and allows you to more easily compare it other costs and its impact on profitability.

THE COST TO CREATE A DOLLAR OF MRR (P)

What does it cost a SaaS company to add $1 of new monthly recurring revenue? Using the typical acronyms:

CAC **(Cost to Acquire a Customer)**
CAC is the total, all-in cost to get one full customer in the door—Sales and Marketing costs, including the fully-loaded salaries of the folks in those divisions, including commissions. The simplest way to compute it is "total Sales and Marketing spend in a month" divided by "total new customers added during that month."

ARPC **(Average Revenue per Customer)**
ARPC is the average monthly-recurring revenue you get from a customer. The simplest way to compute it in aggregate is "total recurring-revenue in a month (MRR)" divided by "total number of customers during that month (N)".

Since it costs CAC dollars to get one more customer who delivers ARPC dollars per month:

p = "the cost to acquire one more dollar of MRR" = CAC / ARPC

PAYBACK PERIOD (P)

The weak* definition of "pay-back period" is "the number of months it takes for a customer's revenue to 'pay back' the cost to acquire that customer." So, if it costs $80 to acquire a customer (CAC) whose MRR is $10 (ARPC), then it takes 8 months (CAC / ARPC) before the customer turns profitable.

You'll notice this is the same formula that we just gave for "the cost to acquire one more dollar of MRR." This is why we named the variable p. Often it's easier to think in terms of "pay-back period," and also easier to find benchmark data for other companies in your industry, size, and age.

COC—THE COST OF CANCELLATIONS

Let's call c your **monthly cancellation rate**, in terms of MRR. So if 4% of your revenue cancels each month, c = 0.04.

Each month, MRR × c dollars will never recur again. You have to replace those dollars for revenue *just to stay level* (forget about growing!). And it costs p to replace each one of those dollars.

* The complete definition of "pay-back period" should also include gross profit margin; if you're curious, see this section of this article about how annual plans can transform your cash-flow (p. 357)

So, the cost, in dollars, cost to replace those cancelled dollars is:

"cancellation replacement cost" = p × MRR × c

We could compute the same thing as a percentage of MRR, rather than as a number of dollars. This simply means dividing by MRR. This yields the metric that is the subject of this article: COC (the Cost Of Cancellations, pronounced *see-oh-see*)—**The *percentage* of our revenue we'll have to spend this month, just to keep from shrinking**:

COC = p × c

THE SURPRISINGLY HIGH COST OF CANCELLATION

Some examples make the utility of this metric clear:

A healthy SaaS business serving SMBs might have a cancellation rate of 3%/mo (c = 0.03), and a marketing pay-back period of 7 months* (p = 7). In this case, COC = 0.21, which means a whopping 21% of its revenue *every month* will be spent *just keeping revenues level.*

That's a tremendous percentage of revenue just to keep from shrinking! That doesn't include the costs of serving those customers (Customer support, SaaS infrastructure, 3% in credit card fees), that doesn't include engineering costs, that doesn't include Sales and Marketing costs to actually *grow* revenues... that's merely to stop shrinkage.

* For example, suppose the cost-per-click on their Google Ads is $2, with a 1% conversion to sale, on an average sale price of $30/mo.

Enterprise SaaS businesses often have a lower monthly cancellation rate but much longer pay-back periods. 1.5% monthly cancellation and 18 month pay-back period means a whopping 27% of revenue is spent replacing cancellations. Are these numbers true in the real world? Yes: Many public SaaS companies have stopped net-growth in terms of absolute number of customers, growing mostly from upgrades by existing customers, because it costs so much to replace the cancellations. (Example: Fastly[1213])

A no-touch SaaS business driven by word-of-mouth marketing might have lower pay-back periods due to efficient acquisition costs, but have higher cancellation rates due to the lack of human touch and poor quality "self-service" marketing channels. I know a prominent SaaS business with a pay-back period of 2 months but a cancellation rate of 5%—that's 10% of revenue to stay even.

That's actually pretty good, compared to the other examples! 5%/mo cancellation means half of their revenue cancels within a year —crazy high!—but the cost of acquisition is so incredibly low that the company is still spending only 10% of revenue to stay even, and of course an additional 10% of revenue spent on marketing allows them to grow at a reasonable clip.

What definitely doesn't work is, for example, a 5% cancellation rate with a 12-month payback period: You burn 60% of revenue to stay even, which means it's almost impossible to grow profitably; in fact you might be shrinking.

LESSON #1: COC CAN BE AS BIG AS OTHER MAJOR EXPENSES

The massive size of COC for most SaaS businesses should be a wake-up call. COC expense can often be as high as R&D or G&A, which

means the business has an unprofitable business model (p. 1087), even once it achieves scale.

A SaaS business must work constantly to reduce COC. Because the definition is so simple, it's obvious that reducing COC means decreasing cancellations and decreasing p, and decreasing p in turn means decreasing CAC and increasing ARPC.

This is a key insight to many SaaS operators, because **typical metrics literature focuses only on reducing cancellation rate, which is only a third of the story.** Furthermore, unless you haven't yet reached product/market fit (p. 335), cancellation rate is often hard to shift compared to reducing CAC (smarter campaigns, optimized landing pages, self-serve sales/on-boarding) or increasing ARPC (using tiered features to segment customers with different budgets and requirements, *a la carte* add-ons for new products or services, price scaling with usage).

For example, my company WP Engine's cancellation rate is under 2% per month. That's low for the hosting sector. Furthermore, when we poll customers who cancel after the first 90 days, the most common reason for cancellation is "project ended." Meaning, it's not something we can affect by changing the product.

Therefore, pay-back period is a smarter place for us to focus in terms of reducing COC. In fact our CAC is also already very low, due to tremendous word-of-mouth that our lovely customers bestow upon us. (And now you see how much we appreciate that!) But maybe we should do more to increase word-of-mouth activity while we continue to optimize our paid advertising campaigns. And what about MRR? We've started selling SSL certificates to customers who want secured sites, which results in incremental revenue (and a better customer experience, because we handle that mess for the customer, including renewing and re-installing the certificates each year).

As another example, take the company above with the 5% cancellation rate that traditional metrics literature would say creates "impossible headwinds for growth," and yet they have a better COC than

a typical enterprise SaaS business, demonstrating that a very good pay-back period can overwhelm a crappy cancellation rate.

LESSON #2: ZERO NET CHURN TRUMPS BOTH CAC AND MRR

The standard SaaS metrics literature does provide a way to combat cancellations: Up-selling existing customers. If you have 2%/mo cancellations, but if on average you *increase* MRR by 1%/mo with things like customers graduating to larger tiers, adding more "seats," buying add-ons, buying premium support, etc., then *effectively* you're only losing 1% of your MRR per month, not 2%.

Thus, c above is not really cancellation rate, but rather "net churn," meaning cancellation rate, plus downgrades, but minus upgrades. The strongest SaaS companies have negative net churn!

You can see the effect of approaching zero net-churn in COC: If c is 0, then COC is 0, which means "getting back to even" costs nothing at all. Phew!

Another fact pops out: In terms of "not shrinking," suddenly p—and therefore CAC and MRR—doesn't matter at all! Of course they *do* matter for cash flow and efficient growth, but at least in the "revenue headwind" sense they fall away.

This highlights the fact that **getting to zero net churn is the strongest thing you can do** in terms of COC. And since we just talked about "cancellation rate" having a floor, that means you *must* develop paths for up-sells.

COC: A NEW STANDARD SAAS METRIC?

Typical SaaS metrics literature characterizes it this way: "As a SaaS company scales, growth rate diminishes, but cancellation rate doesn't. That means it gets harder stay ahead of the headwinds created by cancellation. So you need to work on getting to zero net-churn and, if you don't, you cannot build a large SaaS company, and certainly not a profitable one."

That's true. Casting those words terms of COC is an easier way to measure exactly how big the "headwinds" are, and what it's directly costing you, every month. It allows you to directly measure that phenomenon, which means you can set goals and track progress. It also correctly balances the headwind of cancellation with the tailwind of low CAC, allowing you to make better decisions about where to spend your time to improve the business.

Chapter 122:
Solving the Low-Budget Online Marketing Dilemma

MAXIMUM INVENTORY · NO BENEFIT OF THE DOUBT
WEBSITE TRUMPS ALL · ONLY LARGE EFFECTS
LOW-BUDGET ADVICE ONLY

"No, you're right. 'Eat' is better."

credit 1214

Stop me if you've heard this one:

Your bootstrapped startup is finally off the ground. You're able to spend $6000/mo on AdWords to drive leads. Sure the conversion rates could be better, and sure it's not the best ROI on Earth, but on the balance it's making money.

You don't have a huge budget, but you can plough some of your winnings back into advertising.

You've heard banner ads don't work well, but they're cheap, so you start throwing $600/mo into an ad network and trust your web analytics to tell you whether it's working. The jury is still out, but it's not looking good—there's been only two signups in the first month. Maybe your ads suck?

You've heard affiliate programs can work wonders, so you sign up with an affiliate provider and figure you can afford to pay $50 per signup. A hundred affiliates take the bait, but two months later, half of them haven't sold anything, and most of the others have sold only once. Two are producing five signups a month, and it's only cost you $1000 so far, so it's not all bad, but it's not moving the needle.

So now what? Should you work on optimizing AdWords since they're working, or optimizing banner ads since they're not working? Should you cut off the affiliate program since it's a waste of time, or redouble your efforts in affiliate management? Should you optimize these existing channels or try to find a new, more productive channel?

These aren't hypotheticals—I talked to no fewer than four startups in the past few months in exactly this position, with these advertising channels, with about the same costs. And my own company WP Engine wasn't dissimilar two years ago.

How can you tell what's eventually going to work? Maybe you can't (p. 159).

Here's some tips:

"MAXIMUM INVENTORY" IS THE ANSWER TO "SHOULD I OPTIMIZE?"

A common question is: "Should I spend more time optimizing AdWords? Or find more keywords? Or focus on another marketing campaign completely?" A common—and unacceptable—answer is "It depends."

I approach it as a matter of inventory. In advertising, "inventory" means "all of the available advertising space." In a magazine, that means the total square centimeters of space allocated to the adverts.

In AdWords, that means the maximum clicks you can get across all relevant keywords. If you're the #1 spot for a keyword, you're already getting the maximum number of clicks for that keyword, and therefore you're already at the maximum available inventory. (Of course there's more inventory for other advertisers, but AdWords allows *you* only one slot, so this is the maximum available to *you*.)

So, how much additional inventory is available for you in AdWords? If you're in lower spots for most keywords you care about, or if there are other keywords you're not yet bidding on, there's probably a lot more inventory you could be taking. That implies it's wise to optimize—there's more sales for you there, and it's easier and cheaper to optimize an existing campaign than to start up a new one. You might even experience some cost savings (per signup) as part of your optimization—bonus!

If you think there's 2x or more inventory you could go get, I say go get it.

However, that's not the case for us at WP Engine. At the instant of this writing, we're the #3 spot on AdWords and the #3 organic spot for "WordPress Hosting" which is one of our most important keywords, and #1 for "Managed WordPress Hosting" which is perhaps our most accurate keyword. And traffic on related keywords diminishes quickly and therefore improving on those is even less interesting.

We don't have a lot of inventory left. We believe less than 2x. Therefore, even if we spent tons of money on human analysis and were willing to lose money on every sale, it still wouldn't be an area of significant growth for us.

Once you approach inventory limits, you need to find other campaigns which can double your business (p. 839).

NO BENEFIT OF THE DOUBT

When you try a new campaign and it utterly fails, the temptation is to keep spend low and optimize. You think:

> I might be just a clever turn of phrase (p. 627) or an eye-catching design (p. 853) away from changing that click-through rate from 0.1% to 1.0%, and then I can ramp up spending and have a new advertising channel.
>
> After all, this same messaging worked great on AdWords, so I know it's reasonably well-tested. Something small is probably holding me back.
>
> And I recognize my marketing prowess with this company is nascent, so of *course* campaigns will suck at first but improve with iteration. So let's iterate!

I've found that this line of thought is usually wrong. I don't have data to give you, it's just been my experience.

Thinking about the 100+ campaigns I've been a part of in the past 11 years at Smart Bear and WP Engine, the ones that were bad out of the gate tended to stay bad.

Maybe it's because by the time you get to the third campaign your messaging is pretty good, so an early failure means it's a bad channel

rather than a bad message. Maybe it's because iteration gets you incrementally better but not drastically better[1215] and therefore can't save a fundamentally bad channel. Maybe it's because a failure is an indication that you don't understand the channel at all, a condition that simple iteration won't remedy.

In any case, cut out early and go find another channel.

WEBSITE TRUMPS ALL

Optimizations you do on your website are more valuable than those you do on individual marketing campaigns. The reason is obvious in retrospect: all marketing campaigns lead there! So a 10% improvement in bounce-rate off your pricing page means 10% more revenue across all campaigns: paid, organic, and word-of-mouth.

Landing-page optimizations can also be shared. The exact text can't, but often you can cross-apply a layout that increases click-through rate, a form that captures an email instead of signing up directly, or a message to one customer segment that happens to flow through multiple marketing campaigns.

Often an early-stage startup has large-effect optimizations lurking all over its website. Maybe the testimonials or demo page is missing, or not compelling, or you realize testimonials need to be on all pages, or a long-scrolling, well-designed home page triples click-through rates, or you didn't have a strong call to action, or the pricing page was confusing, or your main hook and product description was turning people away.

ONLY LOOK FOR LARGE EFFECTS

When you're small, almost no data you have is statistically significant. Therefore, most variation you see is due to random fluctuations, not real results, even if you're using one of those tools which supposedly helps you with the statistics.

Here's a quick video I made, that explains why this is the case, including a fun puzzle (see if you can solve the puzzle before the video does):

The punchline is that you should be seeking only large effects, not incremental improvements.

If you're getting one sign-up a day, a "10% improvement" still means one sign-up a day. You need two signups a day. You'll know if you get a change that drastic! Those are the only important changes.

Changes must be drastic, but also based on a theory, not "throwing shit at the wall." Otherwise your A/B tests will be wrong even with a positive result (p. 913).

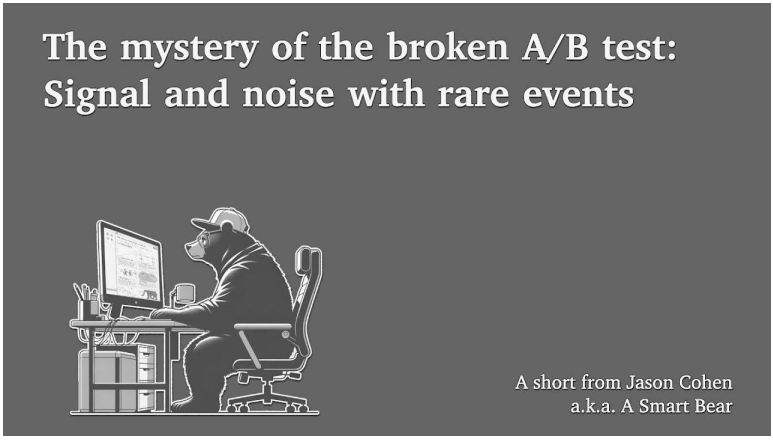


Figure 1: Watch on YouTube[1216]

LOW-BUDGET ADVICE ONLY

Finally, as the title suggests, this is advice for *low budget* online marketing. Large budget marketing is completely different. You try as many channels as possible, as fast as possible. You hire experts in the hope that it will accelerate or de-risk the effort. You find the ones that work quickly, then immediately ramp to full inventory. For channels like affiliate programs where you can't just "write a big check" and make them work, you develop an entire team to extract as much value as possible. The fact that you burn a lot of cash on things that don't work out is *exactly* the trade-off you want.

Accelerated (cash) burning for accelerated learning. Good work if you can get it!

VC-funded companies *should* do this; it's a way to spend money to grow faster. Even bootstrapped companies might be able to do this if they adopt annual pricing (p. 353).

But if you're cash-strapped, use inventory limits to decide whether to flog a channel, get out of bad channels quickly, optimize your website before optimizing channels, and don't get excited by tiny, immeasurable results.

Chapter 123:
Bootstrapped CPC rule of thumb:
ARPU/25

LTV = ARPU X 20 · CAC = LTV / 5 · CAC = ARPU X 4
CONVERSION RATE = 1% · CPC = ARPU / 25
CUSTOM

"I see big changes in your future. Which reminds me, my fee is going up beginning next week."

credit [1217]

In the first year of business, you have no data for decision-making.

Even after the first hundred customers, half of those were serendipitous one-offs, not representative of repeatable, predictable customer acquisition, and the scale of the data isn't statistically significant.

One of the fundamental data-driven questions (but you don't have data) is: **What's the maximum I should bid for CPC (cost-per-click) campaigns like Google AdWords?**

The answer for a funded startup is "Bid as much as possible, to get as many customers—and data!—as you can, as quickly as you can, then rapidly iterate from there in the presence of that data."

That's a smart use of money: To "pay to find out." But what about a bootstrapped, profit-driven business? You don't have that budget, and you're keen on getting a reasonable return on investment reasonably quickly (p. 353).

Here's my way.

(Tune the exact numbers if you disagree with my assumptions!)

LTV = ARPU X 20

ARPU (**A**verage **R**evenue **P**er **U**ser) is the amount you charge the average customer every month, which is typically a mixture of different quantities of customers at different tiers, special add-ons, etc..

LTV (**L**ife-**T**ime **V**alue) is the total amount of money you expect to collect from a customer over their entire tenure. A simple version*

* The correct version also includes multiplying by Gross Profit Margin, i.e. the cost to serve customers, which for SaaS is tech support, server infrastructure, and payment fees. You should include this for a more accurate calculation; small bootstrapped companies often have very high GPMs, so ignoring it for this back-of-the-envelop calculation was simpler.

is `ARPU × [expected months]` meaning the *average* number of months a customer sticks with you.

Some customers cancel in one month, some cancel in a year, some in five years, and some never cancel! So it can be difficult to compute LTV accurately for small companies, and impossible to know for young companies (where five years hasn't elapsed yet to see how many customer stuck it out that long). These are among the reasons that I dislike the LTV metric (p. 1355), but it's common to use it in this context.

If you do have data, the simplistic calculation is `[expected months] = 1/c` where `c` is your monthly cancellation rate.

But since you don't, in my experience (and in a non-scientific survey of some of the 100 startups currently officed at the fabulously Capital Factory co-working space[1218] in Austin), a good **pre-data rule of thumb is 20 months**.

If you have an average customer lifetime smaller than 20 months (i.e. cancellation rate higher than 5%/mo), that's a dangerously high cancellation rate (p. 335) for almost any SaaS business, and you need to focus on addressing the business issues (p. 1191) before acquiring more unsatisfied customers. Use surveys and one-on-ones to try to understand whether it's technical failings, lack of features, missed expectations, bad service, doesn't hit pain points, or what.

A healthy SaaS company will have a higher number of expected months, but at the start you also will have lots of mis-steps with weird early-adopters and non-ICPs (p. 317) where your product is at its worst—least features, least quality, etc—so it's good to assume a low LTV instead of inflating it to where it might be in future.

CAC = LTV / 5

CAC (**C**ost to **A**cquire a **C**ustomer) is your *average* total cost to get a new customer, which includes direct costs (AdWords spend, affiliate payouts, the fees your affiliate system charges to process them) and indirect costs (consultants and your own time). So to compute CAC, take your total costs to acquire new customers and divide by the number of customers you acquired.

In general of course CAC needs to be less than LTV, otherwise it costs so much to get the customer that you will never make money. **A surprising number of startups have CAC > LTV.** Many justify this either by not correctly computing CAC (e.g. ignoring indirect costs) or saying they'll "fix that later" by raising prices or finding other channels of revenue. Others justify by saying they're doing a "land-grab" for customers, and just having a customer at all has intrinsic value.

Profit-seeking bootstrapped companies cannot afford those delusions. Also you need something far stronger than CAC = LTV, because you need to pay for other business expenses and still produce a profit. So how big can CAC be before it's "too big?"

Growing, funded SaaS companies who treat CAC with respect often commonly target `CAC = LTV / 3`.

Back at my second startup IT WatchDogs, my co-founder Gerry Cullen used to say "A third to built it, a third to get rid of it, and a third to keep," meaning a third of revenue goes to pay for hardware/inventory/shipping costs of the sale, a third goes to what I'm calling "CAC" here, and a third for the overhead costs, development costs, and profit.

That's a good model, and I think a bootstrapped company can copy it, but I urge profit-seekers to instead adopt an even more strict model of CAC = LTV / 5. The reason is that at the start you should be able to find (p. 1369) a few efficient ways of acquiring customers, even if those get saturated over time (p. 115).

CAC = ARPU X 4

If you combine the previous two results, you see that the cost to acquire a customer should be no more than four months of revenue.

Another good way to think about it is: "The payback-period for my cost to acquire a customer is four months." Also, ideally you're getting the first month of revenue back immediately (p. 353), so it's really three months of cash-float.

Companies with large budgets to deploy at scale will often be happy with 12 month payback periods; some very high volume businesses like shared hosting will accept 24 or 36 months! But a bootstrapped company's cash-flow won't allow it, even if the math would work in the long run.

CONVERSION RATE = 1%

Conversion Rate is the percentage of visitors to your website who convert to a *paying* customer.

This is another step which in practice should be completely data-driven, segmented by customer type and marketing channel, segmented by landing page, A/B tested (p. 913) and iterated, blah blah blah. But since you don't have data, and you don't have enough visitors to have real ratios, you have to take a swag at this number.

In that same informal survey I ran (p. 71), and bolstered by other formal surveys,[1219] a huge number of bootstrapped SaaS companies report a 1% conversion rate.

Another way of saying the same thing is **"You need 100 visitors to make 1 sale."**

And since you need to incur no more than **CAC** dollars in the making of that sale, you need to incur no more than **CAC/100** dollars in the making of each of those visitors.

And if you're running a CPC campaign, that means you can pay up to CAC/100 dollars per click.

And since CAC is ARPU x 4, we can substitute and get the end result:

$$\text{CPC} = \text{ARPU} / 25$$

So for example if your average customer generates $50/mo, you can spend $2/click.

Indeed, this is a great way to prove one of my main arguments for all bootstrapped companies, which is that **you should charge a lot more than you think**, in part because it enables you to pay quite a lot per click, which enables a wide number of marketing channels, and out-bidding parsimonious competitors whose paltry LTVs preclude them from competitive marketing spend.

CUSTOMIZED

"But my numbers are different!" Of course, but now you have a formula you can plug them into, to arrive at the answer:

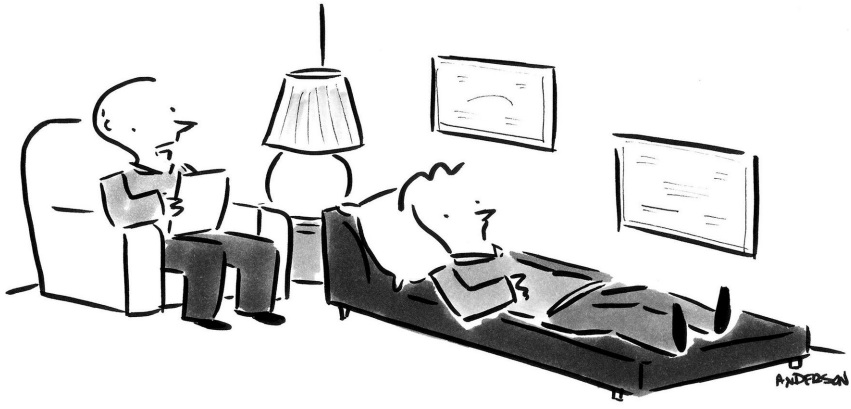
$$\text{CPC} = (\text{ARPU})\frac{r}{5c}$$

Where:

- c = monthly cancellation rate
- r = visitor → purchase conversion rate from the paid marketing source in question

Chapter 124:
Reframing "Freemium" by charging the marketing department

FOIBLES · ADVANTAGES · FREEMIUM AS MARKETING
FURTHER READING



"By the way, I'm having a special this month. Have one breakthrough, get another free.

Seems like every third startup nowadays is using the "Freemium" business model: The lowest service tier is free-forever (so it's easy to get started), with trip-wires thoughtfully arranged in all directions, causing "real users" who are getting "real value" to start paying.

It can work wonderfully of course, but usually it crushes companies, not only because of direct costs, but because the founders didn't realize the business model itself caused them to make incorrect strategic decisions.

FOIBLES OF FREEMIUM

Let's dispose of some misconceptions about what Freemium actually does for you and how much it costs.

Freemium is not customer development

Just imagine how much you'll learn once you have 1,000 real, *active* users of the system: Everything from behavioral statistics (which features are actually used) to democratic product development (voting on which features customers would like to see next).

Trouble is, **those freemium users are not like those who will actually give you money.** Frequently the features that paying customers want don't show up on the free-riders' radar.

Think about it: Almost no freeloader will convert, ever. Even to the lowest tier. If you suddenly started charging merely $1/mo for your service, most would just quit. Why is that? Because the need, the interest-level, the value to that person isn't pressing enough for even a pittance. But those who will pay $10/mo or $100/mo *do* have needs, and it's not just a matter of scale, it's a difference *in kind*.

Counterintuitively, the freemium-tier customers *are not your target customer*, not your "ICP" (p. 317). There are ICPs among those freemium users; your strategy is to use price as a sieve for those ICPs to rise up, giving them as much time as needed for them to realize it. 95% will never rise up.

Your problem is that freemium users outnumber your paying customers 20:1 or even 100:1, so in feedback forums **they drown out the voices of those who actually matter**.

Freemium conversion rates makes marketing expensive

A really good conversion rate for free-to-paid is 4%, like Dropbox.[1221] Awesome for them, but it took a ton of work[1222] to get there, with one of the most successful Freemium products of all time. Normal rates are more like 1%, especially when you remember that most freemium users not only won't convert, but won't even stay active.

I surveyed* a dozen small startups who *don't* use freemium, and on average they see a 1% conversion rate from web traffic to purchase. The conversion rate to a free tier is higher than the conversion rate directly to a paid-tier (it'd better be, right?), but sadly it's not typically that much higher: 3%-5% is normal, 10% at the extreme. But you only get paid on the few percent of those freemium users who convert, which again might be 2%-5% (with outliers like Slack and Spotify which are around 30%), which means your total conversion rate of web visitors to actual money can be many times worse than other startups.**

Which means it costs many times as much as it usually does (p. 1377) for marketing campaigns to achieve the equivalent revenue.

---

* Andy Brice has much more data[1223] that supports my informal observation.

** In OpenView's Product Benchmark data[1224] of 458 respondents, the visitor → freemium conversion rate was 6%, and freemium → paid rate of 5%, for a total visitor → paid rate of 0.3%. Or FirstPageSage's 2025 report,[1225] which showed a larger visitor → freemium rate of 13%, but a smaller freemium → paid rate of 2.6%, for a total visitor → paid rate of again 0.3%.

Annual pre-pay might solve it (p. 353), but maybe not—it's a huge hill to climb.

This essentially takes paid-advertising and almost all other forms of marketing off the table for driving growth in a freemium business, **unless you're willing to take big losses to get things rolling.**

It also means you essentially *have* to build a viral product,[1226] because you can't afford advertising. Getting true viral behavior is very hard—again most companies who attempt this will fail—and even so you need to seed it at the beginning, so you still have the marketing expense problem.

The few companies who were clear winners with this business model also raised tens or hundreds millions in funding, in part to get over this hump.

Freemium tech support is expensive

It's easy to say "We'll just direct everyone to forums," but when people email tech support they want a response. And are you sure other customers want to spend their time being your support team? Is that something you *ought* to expect of your customers?

It's easy to say "We won't provide tech support for the free tier—they'll understand since it's free," but if you really do ignore them they'll be less successful with your tool, which means far less chance of them converting, and less chance they'll evangelize to friends and coworkers. "Spreading the word" is one of the main benefits of Freemium, so losing this might be fatal.

It's easy to say "We'll provide a lesser grade of support for the free tier," but then each rep is making a determination on "how much to help." How will that conversation with the customer actually go? Even setting aside how that will make the customer feel, how will that make the support reps feel, day in and day out, that they're intentionally providing less help to people who need help?

Are you prepared for people who say, "If you help me through this, then I'll pay." Are you hard enough to shut the door in their face, even knowing that in fact they probably still won't pay?

If you care about good support—one of the few true competitive advantages a small startup has (p. 295)—can you really segment who gets treated well and who gets the cold shoulder? Should you really turn your back on the benefits created by great tech support (p. 1503)? Is that conducive to converting free accounts to paid accounts? Is it helping your company's reputation?

ADVANTAGES OF FREEMIUM

Obviously freemium also has important benefits which cannot be denied:

- **Easy to start.** Even a "30-day free trial" or "money-back guarantee" is a much bigger barrier than "free." Getting a web visitor to stop perusing and start using the product is a critical step in any customer acquisition, and you've just diminished the barrier as much as possible.
- **Easy to upsell.** They're already using your tool, so whether it's by special offer, changing the pay scale, or the user simply outgrows the confines of their service tier, there's many ways a person can start giving you money.
- **Stats for selling.** It's awesome marketing to be able to say "Join 10,000 happy users." It's social proof, just like the subscription counter in the corner of a newsletter implicitly saying "if 60,000 people think it's worth their time to read this every week, maybe it's worth yours too." It's also useful when bagging bigger custom-

ers because it proves your system can scale, both in technology and in training new users with minimum effort.
- **Not using the competition.** One more user for you is one fewer user for them. Market share is its own value, especially in zero-sum games.[1227]

So in the face of the positives and negatives, how do you decide whether it's right for you and, if it is, how do you think about it so that you're reaping the benefits while mitigating the costs?

CHARGE FREEMIUM TO THE MARKETING DEPARTMENT

A pattern emerges from those "advantages" bullets: **They are marketing**, as opposed to utility. They are lead-gen, reducing barriers to conversion, and competitive advantage.

Retool your expectations of Freemium: It's a *marketing cost.* It's more expensive than you give it credit for, but it could very well be the best marketing strategy.

So how do you decide whether those costs are worth the benefits? My technique is to "charge the marketing department" exactly like AdWords or any other lead-gen campaign: measuring the total cost of acquiring new paying customers.

The reasoning is: You have a theory that by spending the money to support these freeloaders, you're in fact building an efficient path to real, paying customers. That goal—revenue—could be attained other ways: AdWords, SEO, social media (p. 1027), or any other marketing technique (p. 1459). So just like any marketing campaign, the marketing department should pay, measure the results, and compare the ROI against other methods.

(And make sure the cost is much less than total lifetime revenue (p. 1355)—the equation that many startups fail to achieve, exactly because they don't honestly consider the total cost to acquire.)

How much should you charge the marketing department? Suppose you really were charging those free-loading customers, but only enough to recoup costs, not to be profitable. Amortize those costs—infrastructure and tech support—and come up with an effective "price" for the free tier.

That's what those users *should* be paying (excluding profit), so that's the amount the marketing department needs to "reimburse" the rest of the company.

That said, we agreed there were additional benefits of having free-loading users, such as the value of a competitor *not* having that customer, or the word-of-mouth growth benefit of larger market-share. You can decide what those things are worth, and deduct that from the marketing expense. That is, you could answer the question: "What would you be wiling to pay, per customer, per month, *only* to *prevent* them from being a customer of a competitor?" The answer is probably more than $0 (unless you really have no money, in which case Freemium is probably not the right model for you!), but not infinite either.

You have no excuse to not be measuring this net cost-per-user so that you're running this Freemium program with full knowledge of the costs, and comparing it to other forms of marketing.

FURTHER READING

More great articles on the subject:

- Andy Singleton[1228] on how to "make those cheapskates pay."
- Andrew Chen[1229] on the precise financial model of freemium (spreadsheet included).
- Eric Ries[1230] on three strategies for "free" that still lead to making money.

Chapter 125:
A life-changing challenge guided by Pascal's Wager

"Oh, thank God. Stacy, it's for you."

If the Gartner Group issued a *Recommendation on Behavior of Fortune 5000 Corporations with Respect to the Existence of PBRs (Power Beyond our Reckoning)*, surely they'd back up their ecclesiastical recommendation using a 2×2 diagram where the best place to be located is up and to the right.

Silly, and yet, that's precisely what genius mathematician/physicist/inventor/everything-else Blaise Pascal did hundreds of years ago in his pithy argument for the existence of God:

> Let us weigh the gain and the loss in wagering that God is.
> If you gain, you gain all; if you lose, you lose nothing.
> Wager, then, without hesitation that He is.

Or in modern lingo:

> If you act like God does *not* exist, but it turns out he *does* exist, you're eternal toast.
>
> Whereas if you act like God *does* exist, you can't fail, because either (a) he in fact exists, good on you, and if (b) he doesn't, no biggie.
>
> Thus the only logical strategy is the one where there's no significant downside: To believe in God.

However, if I were drafting the Gartner PowerPoint slide, I would rip off Wikipedia's excessively-technical crowd-sourced analysis (Figure 1).

Actually this isn't a rational argument for the *existence* of God, it's a rational argument for *acting as if* God exists, because it's the bet with the highest expected value (p. 963). (Although this isn't how you should make large bets (p. 867) anyway).

Pascal's Wager can be applied to other things in life.

We react to most things from a position of scant knowledge (p. 193), especially when we're running a startup where essentially

Analysis with decision theory [edit]

The possibilities defined by Pascal's wager can be thought of as a decision under uncertainty with the values of the following decision matrix.

	God exists (G)	God does not exist (¬G)
Belief (B)	+∞ (infinite gain)	–c (finite loss)
Disbelief (¬B)	–∞ (infinite loss)	+c (finite gain)

Given these values, the option of living as if God exists (B) dominates the option of living as if God does not exist (¬B), as long as one assumes a positive probability that God exists. In other words, the expected value gained by choosing B is greater than or equal to that of choosing ¬B.

In fact, according to decision theory, the only value that matters in the above matrix is the +∞ (infinitely positive). Any matrix of the following type (where f_1, f_2, and f_3 are all negative or finite positive numbers) results in (B) as being the only rational decision.[4]

	God exists (G)	God does not exist (¬G)
Belief (B)	+∞	f_1
Disbelief (¬B)	f_2	f_3

credit [232]

Figure 1

every decision is a guess (p. 159). An educated guess, but not *terribly* educated (p. 429).

You have to make assumptions of course, but Pascal points out that some assumptions fare better in all eventualities, whereas other assumptions result in positive outcomes only sometimes. Surely we should choose the behavior that maximizes our expected value as Wikipedia suggests.

Sounds obvious but we don't always do it. For example:

Folks often contact WP Engine tech support from posture of accusation. They're blameless ("I didn't *do* anything!") so we must have screwed something up. Occasionally that's true, but most of the time it's either ignorance or someone else on their team messed it up, or

they themselves actually did mess it up, hoping we'll come to their rescue. (Which we will.)

Now, look. Support people are human beings. Not mechanized automatons[1233] impervious to rude language (p. 1533) and assumption of incompetence. They've also chosen a field in which they help people.

Meditate on that for a second. They've chosen to *help* people.... for a *living*. Is that how you'd define most of your day? Is that not a noble profession? And what thanks do they get, call after call, day in and day out?

So if you open up the conversation from a posture of helplessness or curiosity, they're inclined to help, even if it's your own fault. And if turns out *not* to be your fault, imagine how sympathetic they'll be to your cause—here's the nicest person on Earth, blaming themselves from the get-go, and yet it's *our fault*! Imagine how readily they'll personally fight to remedy the situation.

But that's not what you do when you're on the phone with tech support, is it? No, you're angry and frustrated because something is wrong, and you unfairly take it out on whomever you're talking to, even though they're also your only link to salvation. But then how do they feel about helping you? Of course it's their *job* to help you, and there's metrics and such which hopefully punishes them if they don't do their job. But still... **is that the way to get most out of other people?** Is that the way to live your life (p. 399)?

You can't fail if you assume you're ignorant, that you're missing information, that you're ready to learn, that you need help to understand.

It's just Pascal's Wager again: **Being humble cannot fail; being arrogant can**.

This isn't specific to WP Engine's support, of course. It's everyone, any time, especially if you're communicating over email where emotions are hard to convey and text is misread. Tech support, vetting a new idea (p. 239), arguing with your spouse, or exchanging emails or Tweets with a stranger.

But how often do you act like that?

What would happen if you always acted like that, in every situation? Would you discover you're wrong more often and about more things than you thought? That you misread, or read something that doesn't exist, in-between lines that aren't the lines you thought they were?

And, in the cases where you are in fact correct, perhaps people would respond in a more positive way, where they learn also, and where they go out of their way to make things better? **Could the simple act of being humble be life-altering?**

Try it for a week and just see.

You have nothing to lose and everything to gain by trying.

Like Pascal.

Chapter 126:
Which is better: Many customers at low price-point or few at high price?

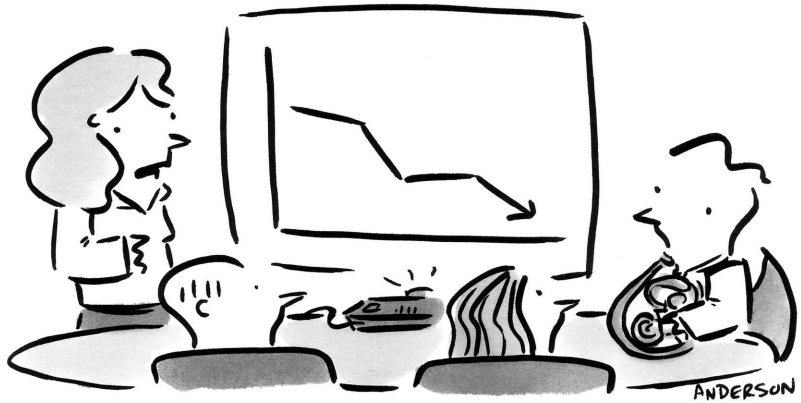
MARKET SIZE · MARKET RISK · TIME · PIVOTABILITY
FOR B · FREEMIUM · DECISION

The results of a serendipitous live experiment were recently published as guest posts on this blog. Sacha demonstrated[1234] the benefits of selling many copies of an eBook at a low price, while Jarrod pointed out[1235] the advantages of higher prices, bringing in more revenue with 1/6th the number of units sold.

The ensuing discussion swirled around the merits of selling more units (i.e. maximizing reach) versus selling more expensive units (i.e. maximizing per-unit profitability). **This is a choice that every startup founder must make**, so I'd like to dig in deeper.

To clarify the discussion, let's use a simpler model:

> Companies A and B both sell products with recurring monthly revenue, and both brought in $10,000 in revenue last month.
>
> Company A has 1,000 customers each paying $10/mo.



credit[1236]

"I think if more french horn were going to work, it would have by now."

Company B has 10 customers each paying $1,000/mo.

Which is better?

Oops, bad question. How about: Which company would you rather own? Or: What primary problem should each company be working to solve? Or: Under what conditions are each of these companies interesting? Or: Which company could raise money more easily?

Let's focus on just one question: **For which company would be easier to raise money?**

Wait! That's shitty! Why the obsession with raising money, what if you don't want a huge company, what if you want to bootstrap, don't you know raising money isn't a measure of correctness or success, ...

I agree! But "raise money or not" is also a decision everyone must make, and it turns out that exploring that question will end up answering all the other ones. So let's play!

MARKET SIZE

Suppose the total addressable market is small. In that case, A can't keep growing forever, so its revenue is limited, which is a bad spot. B can extract more money from the limited pool of customers, so that's better. Except, of course, investors don't like small markets!

In a large market, B isn't necessarily bad, but A shows far more potential. Over time companies at small price points are able to increase prices and otherwise extract more money from various customer segments, which means A has a bigger revenue potential.

Perhaps most importantly, A demonstrates that there *is a large market at all*. If you've already found 1,000 customers, there's 10,000, and likely 100,000. If you've only found 10, there *might* be 10,000 out there, but if so, you don't have supporting evidence. Riskier.

Speaking of risk...

MARKET RISK

Many companies die because they can't find enough people to pay. Many more die that way than die because the product sucks or doesn't have enough features or because they don't have a staff designer.

There's a million variables—can you locate potential customers, can you bring them to your website, can you get them to read and click, can you get them to sign up, can you get them to agree to your price. A million variables means it's hard to get it right.

Therefore, an investor is always impressed with a company like A who has made irrefutable progress on this particular front. Having 1,000 people paying you *any* amount of money whatsoever goes a long way. It's a lot harder to get 1,000 paying customers than to add

three features, because the latter is a matter of time and money (p. 1445) whereas the former is largely out of your control (p. 1541).

Getting 10 people to pay you—even a large amount—is actually not that hard. If a co-founder has a Rolodex in the industry—extremely common—then it would be surprising *not* to find 10 people. That doesn't prove you have a repeatable, scalable method for finding customers, nor that there *are* a lot more potential customers out there.

Market risk is most startups' biggest risk. One interesting way of reducing that risk is to build a company like B where you just don't need to sell very much to achieve your goals. That's awesome because the risk is low when the bar is low. That's not intended as an "insult"—in fact I believe far more companies *should* have this attitude.

TIME HEALS MANY WOUNDS (BUT NOT ALL)

Over the time scale of "years," you can count on certain trends.

For example, **the average cost of customer acquisition diminishes**. Why? Because you get organized around marketing metrics, because your campaigns get optimized, because your landing pages and drip campaigns become stronger, because word of mouth produces sales "for free," and so forth.

Another is that **average revenue per customer increases.** Why? Because new pricing tiers better segment customers, prices go up as reputation grows, you create add-on products and services, you create new revenue through business development, and so forth.

What's *not*true is that you always unlock big growth drivers.Indeed, many companies get stuck at a certain growth rate which, while positive, eats too much money during its slow crawl to cash-flow-positiveness, and by the same math doesn't generate interesting

profits after that. Once profitable, at least that sort of company is creating jobs and still could unlock something someday, but of course an investor in general isn't interested in that outcome.

So back to our two companies. Company A has demonstrated that some growth is possible, and where there's 1,000 customers from a shoestring budget there's likely several other growth drivers out there; anyway, one is unlocked. Which is more than you can say for B. So, along one of the dimensions which doesn't automatically improve with time, A wins.

That's why, even if A isn't doing well in other areas, that's not as important. Suppose you argue that $10/mo isn't enough money to be interesting—perhaps, but average revenue increases, so that's not a long-term problem. Suppose you discover that it costs $60 to acquire a new $10/mo customer which is too much to be sustainable—perhaps, but that cost diminishes over time, so it's not a long-term problem.

Investors are of course more interested in where you could be in two years than where you are right now. They're more worried about the problems which don't naturally get corrected over time.

PIVOTABILITY

Nowadays everyone agrees that it's both likely and healthy for an early-stage startup to be on the lookout for an intelligent pivot.

Actually, more than "on the lookout," you should be actively probing the market, which means interviewing customers and non-customers alike, attending industry events to have real conversations (not quipping to each other on Twitter), exploring the metrics of your website, your marketing, and product features, and so on.

One of the most common answers to "what made you successful" is "we decided to stop X and do Y." Therefore, actively collecting the

data on what's *actually* happening, what customers *actually* will pay for, where the valuable hole in the market *actually* is—this is one of the most valuable things you can do, and the company which does it best is increasing its chance of success.

Given no other information about the companies, company A clearly has access to far more market data. They have 100x the quantity and range of customers to interview and analyze. They probably have a correspondingly large amount of website traffic to mine. They can subdivide their user population and try four ideas at once, iterating quicker to better information.

Lean Startup tells us that the speed at which theories can be tested is directly proportional to learning; the company who can do that faster and more accurately has a significant advantage.

I posit that this is true regardless of whether you're taking investment.

A FLURRY OF ARGUMENTS IN FAVOR OF B

So it's clear that in general an investor will prefer A to B. But B is preferable in many cases, so let's even the score.

If the cost of support is high, A will kill profitability and B wins.

If the cost of customer acquisition is 10x the monthly revenue or monthly revenue is 1/100th of where it should be to sustain the operations of the company, then the argument of "it gets better over time" doesn't work, because although it gets incrementally better, it's hard to justify orders of magnitude of improvement.

If the human cost of scaling A is higher than B, then at scale B might be much more profitable.

If you're keeping the company small, it's almost always cheaper and more fun to run it like B. You spend less on marketing/advertising/acquisition. Less time training customers. You have more time to make customers love you forever and therefore less churn and a happier general existence. In product development you have the delightful job of serving handful people with homogeneous needs rather than appeasing the disparate needs of thousands people who can't agree on anything. Pretty much everything about it is nicer!

If the market is small, it's hard to get more than a few customers, so you need a business model like B that extracts the most amount of money from the limited available pool.

BUT "FREEMIUM" IS NOT COMPANY A

I often see founders and investors alike using many of the above arguments to argue why a company with 100,000 free users is more valuable than a company with 1,000 paying customers. I disagree.

While it's true that the *potential* for the company with vast numbers of freebie customers is indeed there, there's just too many examples of startups with great products, great marketing, huge growth, large customer bases, where they just could not convert enough of the freebies to paid, and even after conversion, not paying enough.

Of course if there *is* a conversion rate, you can start applying the above logic again. Conversions rates increase over time, etc., so as long as the absolute number of paying numbers is interesting and the growth rate is large, you're back to good. Better than good, in fact, because you have more levers to play with in terms of increasing conversions, offering different products, pivoting, etc..

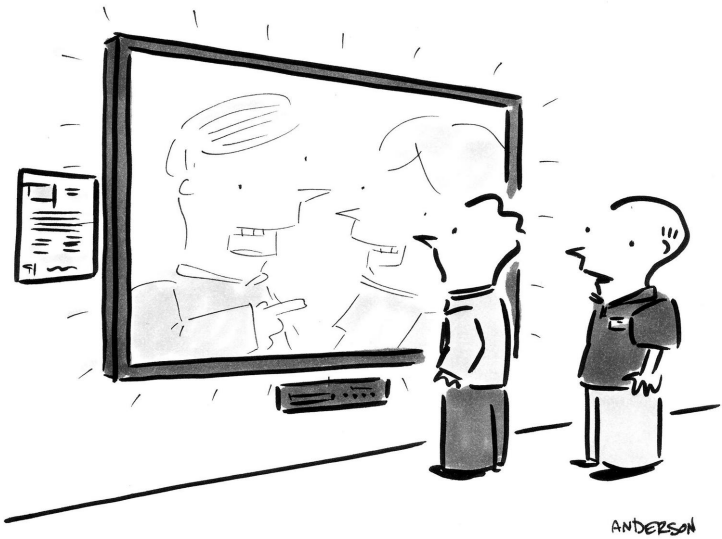
WHICH IS RIGHT FOR YOU?

Hopefully the detail above should be sufficient for you to decide which is appropriate for you.

If I had to boil it down to a sentence it would be:

> If you want happiness and fulfillment from a small company, strive for B; if you want to maximize growth, influence, and financial value, strive for A.

Chapter 127:
Intense Asymmetry and Self-Flagellation



"Only the picture is high definition. The shows are as bad as always."

credit 1237

Erica Douglass[1238] bootstrapped her company and sold it for a million dollars while her friends were switching majors for the third time or having their souls crushed at some entry-level job. She's successful by any definition.

But a few short months of super-charging our marketing efforts at WP Engine was all it took for her to start turning herself inside out[1239] about her "lack of success," and by implication, lack of ability and self-worth:

> WP Engine is killing it right now. They're growing faster than my hosting company did. "Damn," I thought, "I must have really sucked at growing my business." And the downward spiral arrived, ready for me to step on.

All driven people do this to themselves, and it has to stop.

This is not about Erica; she's the example only because she has the strength and audacity to admit it online. **This is about you and me who beat ourselves up** with similar words.

It's tempting to attack her definition of "success" which is apparently things like "growth rate" and "gross profit margin" as opposed to things like "happiness" or "fulfillment" or "loving the state of being alive." I could make this case—that her definition is wrong—and I'd be right, but it wouldn't help you, me, or Erica, because this is not an argument that type-A obsessive people respond to. Even when we know it's true intellectually, our emotions don't respond to that. If they did, economists would actually be right about everything.

Better would be to point out her conveniently gerrymandered criteria, choosing to value "growth rate" while not valuing "monthly cash loss," or valuing "gross profit margin" while not valuing "percent ownership." In fact, you could credibly argue that she's *intentionally selecting* only those elements which will cause her to feel inadequate, ignoring those to the contrary. Which makes me believe **her "explanation" is a rationalization rather than a cause.** The job now is to find the true source.

Well there *is* a source, well-known and backed by studies, called Impostor Syndrome (p. 457). That link is an article from two years ago where I expose myself and others for having exactly this irrational failing.

But maybe you're still not satisfied, because **this is *still* a bunch of touchy-feely bullshit**, and after all **there's nothing wrong with defining success by quantities of money and power** and the speed at which we accumulate them.

Except even if you use that definition of success, her (our!) perspective is *still* wrong, and not for the reason she gave in her article (which was about how hard other people work compared to her, which isn't true anyway because she works all the time too, as my inbox proves.)

To see why, let's wind the clock back to 2002 when Erica started her hosting company, Simpli. And let's suppose I started WP Engine at the same time. (WordPress didn't yet exist in the form and glory it is today, so let's give my company a extra boost in this thought experiment and say that WordPress is around v2.5, circa 2008.)

According to Erica's own criteria, let's see who wins.

WordPress has impressive traction for an open-source tool, but it's orders of magnitude less prevalent than it is today, so I can't find customers in droves like we now do. And there isn't a well-organized panoply of world-wide WordCamps so it's hard to reach out to those potential customers. And I'm not a hardware geek like Erica, and "cloud hosting" hasn't been invented yet, so I have to build, configure, and maintain hardware, which means false-starts and legacy configurations and other expense and waste that Erica would never incur. And I'm in Austin, not San Jose, where there's 100x fewer customers I could land at meetups. And there's no Twitter or Facebook (or even a strong culture of blogging) to enable my few zealous fans to spread the word. And I didn't have several startup successes under my belt and moneyed contacts earned from blogging and speaking, so I can't raise money like we did. And high-tech-America was in post-bubble recession where companies were saving money everywhere they could, certainly not paying a premium for WordPress-specific hosting.

So it's clear how this would have gone down. Either a complete fail or, more likely, *painfully* slow growth, crawling along while taking the same amount of time as the day job I also had to hold down yet generating 1/10th the income, trying to make myself feel better by erroneously claiming I'm profitable (p. 371) and erroneously claiming I have Product/Market Fit (p. 335), eventually realizing this is a cope, culminating in blog post about how inadequate I feel next to hosting companies like Erica's who bootstrapped to success with such apparent ease.

This is what physicists call a "time-asymmetry."

A physical experiment conducted in one location will yield identical results if we move it to another location (provided we move all other influences and pieces along with it). Technically we say that "the laws of physics are symmetrical with respect to translation."

This is more important than it first sounds. Another symmetry is "constant linear motion"—an experiment in a train moving in a straight line at constant speed is indistinguishable from one done on a train station platform. It is *just* this fact, combined with the experimentally-determined mind-bending fact that light always appears to travel at the same speed regardless of the motion of the observer, that gives rise to Einstein's special relativity and explains all electrodynamic theory. This particular symmetry is responsible for half of modern physics!

It's not always obvious where physics is and isn't symmetrical. For example, it *is* symmetrical if you rotate the experiment around a fixed angle, but it is *not* symmetrical if the experiment is being *continuously* rotated (because now there's a centrifugal force). It's almost symmetrical with respect to a mirror, but not quite!

Well it's obvious **where two businesses are symmetrical: Nowhere!** The example above demonstrates how very asymmetrical businesses are in time. In fact they're asymmetrical in everything—location, the market they serve, the features they have, the founding team, their natural talents and gaps (p. 569), the investors (if any), the amount of money raised (if any), the number of social media followers and everything else. If there were symmetries—or anything even

close to a symmetry—we could use that to predict success and failure, at least a little bit. Which even professional investors can only do a small fraction of the time—so infrequently in fact it's hard to conclude they "predicted" anything at all.

But **humans expect, seek, and even invent symmetry.** We crave and seek similarities and patterns and connections (p. 1487). We describe companies that way, e.g. "We're Netflix for art"[1240]—implying a symmetry in target market (i.e. changing the market from videos to art but keeping the website structure, the pricing structure, the delivery structure, etc., ought to result in similar success).

This might be a pithy way to summarize a business, but in detail and execution and route to market and profit margins and opportunity and competition and technology and leveraging your strengths while avoiding your weaknesses (p. 543), and ... well, *everything* ... each company is a new challenge, with new variables, and unpredictable (p. 193).

It's healthy to use (apparent) symmetry to spark an idea, e.g. "That's a creative marketing campaign, maybe we could try something similar."

It's unhealthy to see symmetry where it doesn't exist, then build a straw man to fuel self-loathing. Like Erica. And me. And almost every "successful" person I know.

Erica isn't special because she has misplaced feelings, she's special because she's willing to stand up and admit it in public, in order to help everyone suffering under the same delusion.

I *would* say "stop comparing yourself to others and just be as good as you can," but high-achievers like you and me could never accept that advice, even if it *is* wise, because we're physiologically incapable of not measuring ourselves by impossible yardsticks we invented and, because of Impostor Syndrome, **by definition we always come up short.**

Instead, how about this:

Stop comparing yourself to something that is *objectively incomparable*.

That's just irrational.

Chapter 128:
What is the value of one hour of a startup founder's time?

IMPLICATIONS

"Due to recent economic conditions, picture worth has dropped to an all time low of 842 words."

credit 1241

Almost no startup founder values her time properly.

Consultants know exactly what their time is worth: their hourly rate. As they say, it's how much "the market will bear." (Or "how much the consultant dares to charge.") When a consultant intentionally doesn't work for an hour—whether to be with family or to work on a new startup or to take a nap—they're giving up an hour of earnings.

If being a consultant is your goal, this is indeed how you should value your time (although beware the traps of that business model (p. 701)). But when you're in a startup, the math is completely different.

Your time is $1000/hour, and you need to act accordingly.

Here's why:

Let's suppose you are a consultant who normally charges $150/hour, and you stumble upon a weird client who asks for the following terms:

> "We agree your time is worth $150/hour. However, we can't pay you for four years, at which time we will pay you in one lump sum."

How much should you *increase* your hourly rate to make these terms worthwhile?

It definitely must be increased, not just because of the 4% interest you could be making, but because you can't live off money you don't have, which means you'll need other work too. You must demand a hefty premium to make this inconvenience worthwhile.

But real consultants are screaming that none of this is the biggest problem with this proposal. The problem is: What if this client goes out of business in the next four years and *doesn't pay you at all?*

Supposing this client is an early-stage startup—even if funded—the *most likely* result is that they stiff you! Because they're dead. Let's sup-

pose for the sake of rhetoric there's a 25% chance the company will exist in four years *and* pay their bill.

Like gambling in Vegas, the steeper the odds, the bigger the winnings if you beat the odds. You might think you need to charge $150 ÷ 0.25 = $600 per hour to account for the risk.

But in fact, you need to charge far more! This formula merely brings you back to even! To see why, suppose you divided your time between four companies, all operating on these terms. Chances are all but one would fold, and that one would pay you 4x your hourly rate. But that just brings you back to the same place you'd be if you just charged $150/hour on your standard terms—without the risk, and without waiting four years to get the cash.

So you really need to charge more like $1000/hr in this scenario.

Of course **this isn't hypothetical**, this is *exactly* the terms you're accepting for yourself when you create a startup. The risk is high, so the potential financial rewards must outweigh the risk, which means **you have to value your time at least $1000 per hour**, *not* at your $150/hour consultant rate because of platitudes like "my time is worth what the market will bear."

WHAT DOES THIS MEAN FOR YOUR DAILY LIFE?

It means you don't have time for projects[1242] that have the potential only for small, incremental results.

It means a personal assistant is worth the money.

It means you should delegate far more than you think (p. 981).

It means you have to stick to the rules of finding PMF (p. 9), like talking to customers (p. 239) rather than assuming you know what to build, or working on marketing and sales at least as much as the product.

It means you spend the money on a bookkeeper and CPA instead of messing with receipts, Quickbooks, and taxes. And on help at home with cleaning and chores.

It means you should focus on building things that are even more valuable than money (p. 1445), that generate customer Love (p. 275).

It means you have to work harder (p. 1547) than is healthy.

It means you should obsess about your productivity (p. 925).

It means you have to be smart at prioritization (p. 221), and ensure you're doing it (p. 1065).

In short, it means you need to stay in the intersection of your zone of maximum excellence with the zone of what the absolute most important and urgent things the business needs done.

Because at $1000/hr, you can't afford to do anything else.

Chapter 129:
Hiring Employee #1

It's a big decision to make your first hire, because what you're really deciding is whether you want to keep a lifestyle business (p. 1547) or attempt to "cross the chasm" and maybe even get rich (p. 45).

Assuming you really are in the market for another pair of hands to screw stuff up worse than you already are, the question is how to acquire resumes, how to pare them down, and how to identify someone who is going to work well in your company.

There's already a lot of great advice about hiring at little startups. **Before I give you mine, here are some of my favorite articles**, in no particular order:

I'm not going rehash those or attempt a "complete guide to hiring."

But I do have some fresh advice you might not have seen before:

If a person just left IBM, is she a good fit for your startup?

If she left because she couldn't stand the crushing bureaucracy, the tolerance of incompetence, and the lack of any visibility into what customers actually wanted, then she sounds like a person ready for a startup.

Or therapy.

credit[1243]

"So, where do you see yourself in ten minutes?"

On the other hand, if during the interview she asks how often you do performance reviews, that means she doesn't understand the startup culture. If she says "I thrive in environments with clear requirements, written expectations, and defined processes," run away as fast as your little legs can carry you. (Sorry, too many recent readings of *Tikki Tikki Tembo*.)

Startups are chaotic, rules change, and there is no "job description." It's better to make a strong decision that turns out wrong,[1244] and admit it, than to plan ahead[1245] or wait for instructions. Potential earnings (e.g. stock, performance bonuses) are preferred to guaranteed earnings (e.g. salary, benefits).

You already live by this Code of Turmoil because you're the entrepreneur; you have no choice. But normal people do have a choice and most people abhor chaos. Big companies don't behave this way (p. 1495), and most people are accustomed to working for big companies.

You have to hire someone comfy with the bedlam of startup life.

"We offer a competitive salary, generous vacation time, opportunity for advancement, and, apparently, several high value Pokemon have been found around the office."

credit[1246]

You're not just hiring any old programmer or salesman, you're hiring employee #1. This person helps set the culture of the company. This person has to mesh with your personality 100%. You're going to be putting in long hours together—if they don't get your jokes, it's not going to work.

So why wait until the interview to see whether your personalities mesh? Put it right in the job description.

Be funny, reflect your personality, reflect the uniqueness of your company. See the jobs page at WP Engine[1247] for a bunch of examples—everything from detailing our culture ("Being transparent about our strengths and weaknesses wins us sales") to attitude on writing awesome code ("You think using a profiler is fun, like a treasure hunt") to treating customers ("Whether or not you sleep at night is directly proportional to whether you've made something thrilled or pissed off that day").

You should see the results in the cover letters. If after a job posting like that the person is still sending the generic bullshit cover letter, you know they're not for you. If they respond in kind, good sign.

And anyway, one day you actually might need them to change those pellets, and then you've got it in writing!

On young startups using recruiters, Bryan Menell[1248] sums it up nicely:

> "If you find yourself wanting to hire a recruiter, hit yourself in the head with a frying pan until the feeling goes away."

You need to hire an absolute superstar, and recruiters are not in the business of helping you find superstars.

In fact, their incentives are exactly opposite yours. Here's why.

Recruiters are like real estate salesmen: They make money when you hire someone. They make *the same amount of money* whether it takes you four days or four months to find that someone. So every day that passes, every additional resume you request, every additional interview you set up, the recruiter is making less and less money per hour.

In fact, there's a floor that the recruiter can't go below, so the more you take your time to find the right person the more they'll push you to settle for someone you've already rejected.

The exception is a recruiter who works by the hour rather than for a hiring bounty. These are hard to find but they do exist. I've had luck only in this case.

Think about your own resume. Is there anything on there that qualifies you to run your own company? Not just "experience" generically but really relevant knowledge? I'll bet there's very little. But it doesn't matter, right?

Right, so it doesn't matter with your first few employees either.

Resumes are useful only as talking points. That is, when you have a candidate on the phone, you can use the resume to ask about previous experience, test their knowledge of technologies they claim to have,

etc. Resumes are conversation-starters, but they imply *nothing* about whether the person is right for you.

One particularly useful trick with resumes is to dig deep on a detail. Pick the weirdest technology in the list, or pick on one bullet point they listed two jobs ago that seems a little odd to you. Then go deep. Don't let them say "It's been a while"—if they can't talk about it, how can they claim it's experience they're bringing along?

I don't care if this person is going to spend 60 hours a week writing inscrutable code that only a Ruby interpreter could love. I don't care if the job description is "sit in that corner and work multi-variate differential equations." Everyone has to be able to communicate clearly.

In a modern startup everyone will be writing blog entries, twittering, facebooking, and **God only knows what the hell other new Goddamn technology is coming next.** But whatever it is you can bet it will require good communication skills.

In a small startup there's no layer separating employees from customers. Everyone talks to everyone. You can't have your company represented by someone who can't be trusted with a customer. In fact, everyone needs to be able to not just talk to customers, but even *sell* them. Remember, tech support *is* sales (p. 1503)!

In a small startup everyone has to understand each other's nuances. There's enough crap you're having to figure out without also having to decipher an email. There's enough about your business you don't understand without having to understand garbage sentence fragments in a README file.

Therefore, some part of the interview process has to include free-form writing. In fact, there's a particularly useful time for that....

When you post a job listing—especially on large-scale sites like Monster or Craig's List—expect a *torrent* of resumes. It's not unusual to get 100 in a day. You need a time-efficient system for winnowing them down to a small handful worthy of an interview.

Screening resumes is not an option, because resumes are useless. Besides, you don't have time to read hundreds of resumes.

Instead, prepare an email template that asks the applicant to write a few paragraphs on a few topics. For example:

> Thanks for sending us your resume. The next step in our hiring process is for you to write a few paragraphs on each of the following topics. Please reply to this email address with your response:
>
> 1. Why do you want to work at [company]?
> 2. Describe a situation in your work-life where you failed.
> 3. Describe a time when you accomplished something you thought was impossible. (Can be work-related or personal)
>
> Thanks for your interest in [company] and I hope to hear from you soon.

Here's what happens: First, most people never respond. Good riddance! Second, you'll get lazy-ass responses like "I want to work at your company because I saw you are hiring" and ludicrous answers like "I have never failed at anything."

Resist the temptation to reply with, "You just did." That's what assholes do.[1249]

Maybe 10% of the respondents will actually answer the questions, and you'll know in two minutes whether this person can communicate and, yes, even whether they seem fun, intelligent, or interesting.

One exception to this rule: If the cover-letter is truly wonderful,[1250] that's a rare, great sign and you can probably skip right to the phone interview.

The rule of thumb is that it takes 3-6 months to hire a really good person. Why so long?

- Good people are rare, so it takes a while to dig them up. Like truffles. Or weeds.
- Good people won't change jobs more often than once a year—probably more like every 3-4 years, especially if their employer appreciates their abilities and compensates them accordingly. So you have to find this person in their "once every three years" window.
- Good people gets lots of good job offers (yes, even in this economy) so when you do find one and give them the writing test and then the phone interview and then the in-person interview and then discuss compensation and then provide a formal written offer... there's a good chance they just accepted an awesome offer somewhere else. (This happened to me all the time at Smart Bear. It's happening now at WP Engine.[1251])

This means if you start hiring when you *really need* someone, that's too late. You'll be "in need" for months.

This means you need to be hiring constantly.

So how do you "hire constantly" without being drowned in resumes and interviews? The answer comes from another attribute of good people:

- Good people choose where they want to work, not vice versa. They hear about a cool company, and when they're interested in new work, they call you.

Your company has to be a place good people will seek, not where you have to go fishing. How do you manage that, especially when you're small? Ideas:

- Develop your blog/Twitter so you have a steady stream of eyeballs from people who like you.
- Attend local meet-ups and user groups. Meet the woman who runs the group—she knows everyone worth knowing.

- Sponsor a meet-up at your office. Don't have an office? Co-sponsor with someone who does, like another company or a co-working place. (OtherInbox[1252] is a great example of this; they sponsor the monthly Austin on Rails[1253] user group and the annual Lone Star Ruby Conference,[1254] and as a result all the best Ruby developers in Austin already want to work for OtherInbox.)
- Ask your friends for resumes of people they didn't hire but who they liked. That is, people who are good but just weren't a fit for that company.
- Try to get your "Jobs" page to rank well in local-only search. So e.g. "java programmer job in austin tx," not something impossible like "java programmer."
- Take everyone you know to lunch periodically and ask if they know of a candidate. Yes you can ask them by email but often being in-person brings out more information. Or maybe one of them will be interested himself. (That's happened to me a few times.)

You're hiring a friend, a trusted partner, someone you'll be spending 10 hours a day with for the foreseeable future.

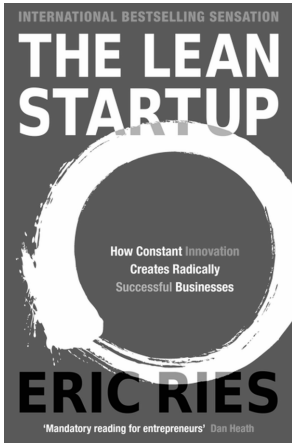
You're not hiring a Systems Engineer III for IBM or a Senior Regional Sales Manager for Dell. The "rules" of HR don't apply to you (except the law).

Think of it more like getting married than hiring an underling.

Going with your gut is not wrong.

Chapter 130:
Ballad of The Lean Startup

With IMVU not growing
And revenue not flowing
And with piles of cash monthly burning,
They asked
"Is it a mess,
Or is it progress?"
Then measured by validated learning.

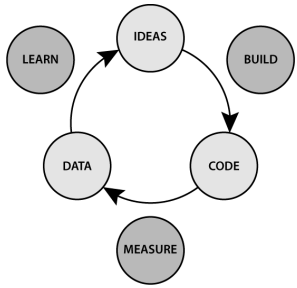


With their customers confused
Eric's ego was bruised—
All his basic assumptions were nixed.
They had to unload
All that beautiful code
Before the product could ever be fixed.

"What's the last six months for,
Tossing my code out the door?"
Asked Eric all in a tizzy.

But he got his first taste
How to eliminate waste,
And make progress instead of just being busy.

You must build and release
(So says Eric Ries)
Then validate by query, wallet, and churn.
Ruthlessly measure
Though it gives you no pleasure—
That's the only true way to learn.

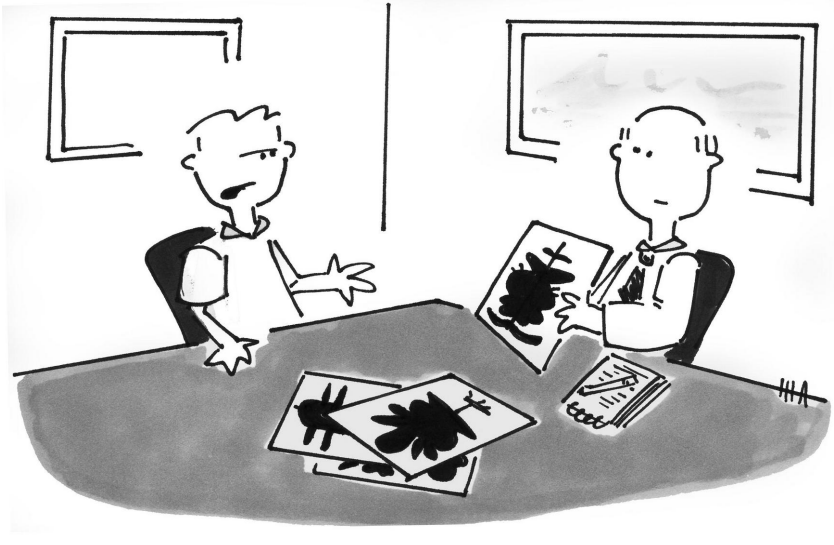


credit[1255]

So don't fly blind
Or make up your mind
Stuck to ideas like a Nazi guards soup.
Relax your tight ass
And gather data en masse
To reduce total time through the loop.

Congrats, Eric,[1256] *on the success of your book.*[1257] *It's well-deserved.*

Chapter 131:
When being an “expert” is harmful



"I told you! It's an inkblot! They're *all* inkblots!"

In a recent Capital Factory[1259]* all-hands discussion, one of the founders started a question with a well-worn preamble:

> "I talked to a bunch of the mentors and they all told me the same thing about pricing, but I'm telling you, they're wrong. **I know our industry**, I know how our customers think, and in our industry ..."

What followed was well-reasoned and sensible. Since none of the mentors have specific expertise in the industry in question, it was impossible to argue.

So rather than argue, I just asked:

> "OK, so when you talked to the last dozen potential customers and proposed the pricing scheme you just described, you're telling me they all said, 'Heck yes'?"
>
> "Well, I didn't actually *ask* them, no."
>
> "Why not?"
>
> "Because I know what they're going to say."
>
> "Great! So, next week you're going to a convention where you'll talk to dozens of new potential customers. Do me a favor —humor me!—and include your pricing scheme in the pitch. I'm sure you're right and they'll be thrilled; since you're so certain, it will be easy. In fact, it will strengthen your pitch because it will match their expectations and therefore mitigate any worry that you don't 'get it.'"
>
> "OK, I will!"

I could already see the self-satisfaction on her face. She *knew* she'd have a great "I told you so" moment next week, and in fact I was equally sure she'd have that moment! She *is* the expert, I'm not. Case closed.

---

* Startup incubator in Austin, Texas

Of course (you know what's coming), it turns out she was wrong. And **whenever an assumption is kicked out from under you, that's when you learn the most.**

The following week she sent me this email. The conversation above was my best recollection, but the following is a direct quote (with my emphasis):

> Ever since accidentally stumbling upon lean startup 1+ years ago, I've struggled to implement the principles correctly. Somehow my version of "lean" customer discovery involved hour long phone calls, relationship-building networking meetings, vague answers to improperly formulated questions...
>
> **In the past week of quick phone calls to vendors, I've learned more about this market than I did in the past year.*** I also got a good feel for when I no longer needed to do further discovery.

We're all plagued by this defect of human nature—thinking we know more than we do—which then causes us to miss opportunities to actually *learn* something. **I still struggle**—in every customer call I have to consciously restrain myself from pitching and instead ask questions, and really try to understand what *they* mean instead of mapping their words onto what I want them to say.

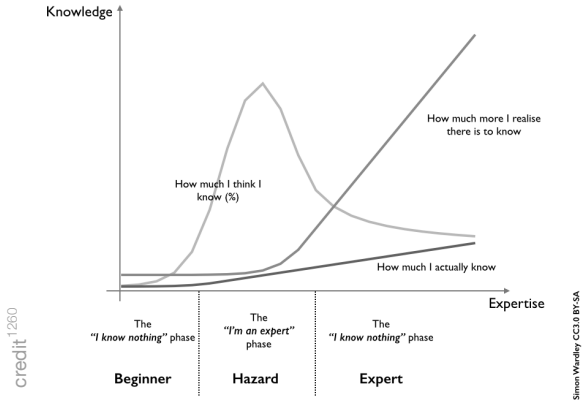
The worst is when you're an "expert" (p. 1529) because then you're even *less* likely to challenge your assumptions.

As an "expert" you've devised your own laws about what makes your market different from other markets, and what makes your company unique. Even with prior experience, this knowledge based largely on *feeling*, not *fact*.

When I say this to "experts," their first reaction is (of course) defensive. "But I have 15 years of experience selling into the financial services sector; I know what makes them tick." Shoot, I used this excuse

---

* If you similarly need guidance on how to do customer discovery better, here's my method (p. 239).



"P.S. Just in case, the above is meant to be a joke and a social commentary on the hubris of self declared experts. It's based upon precisely nothing. Just because someone draws a graph, doesn't make it real." —Simon Wardley, 2008

myself recently: "I built a company and forged dozens of customer relationships in the software development tools sector; I know exactly how to sell into that market."

This is wrong for a number of reasons.

First, markets change rapidly. You can't rely on five-year-old information about your potential customers—even the stodgy big-company ones but especially the mass-market consumer ones.

Non-technical people now employ technology (iPhones, Facebook). Industries built around control of information are now out of control (real estate, publishing). Methods of reaching consumers change every year (compare SEO or AdWords strategies from 2003 and 2010).*

* *Editor's Note from 2024:* In 2003 with Smart Bear I was often the only bidder for a keyword, and paid $0.05/click, and therefore I was able to use AdWords to bootstrap that company. By 2010, when I started WP Engine, AdWords were too expensive to start with anything but long-tail phrases, and only become a staple of marketing when we could afford it, and even then was always one of the lowest-ROI marketing activities. Today in 2024, most bootstrappers say it's prohibitively

Expectations of what software should do, or what it should integrate with, or whether it's OK for data to live on your laptop or someone else's server, or whether it needs to be accessible from a cellphone—it's all changing, all the time.

Ironically your 10-15 years of experience "in the field" **might be clouding your judgment**. Some of that experience is invaluable—so much so that it's an unfair advantage (p. 1475) and something you really should leverage (p. 543). But your outmoded ideas prevent you from addressing the market *as it is today*, allowing a competitor to beat you with **innovative advances they achieve exactly because they're not shackled by old ideas.**

Another way your experience can hamper you is that **selling different products into the same industry is different.**

For example, I talked to an entrepreneur with years of experience selling a standard medical device to doctors. He has an idea for a new software package for managing an expensive, time-consuming aspect of practice-management. Of course his Rolodex gives him an lovely advantage—he can bounce ideas off potential customers and line up ten alpha testers before writing a line of code.

But he hadn't done that. He told me that he's been selling to doctors for years. I asked whether it was OK to install new software on their computer; he didn't know because he was selling hardware before. I asked whether it was OK to depend on an Internet connection inside a secured hospital; he said "probably" but he'd never asked. I wondered what they would pay for this software; he said they paid a lot for this medical device so it should be easy to get lots of money for this software. I doubted that the budget for front-office software has any relation to that for devices; he hadn't thought of that. I asked how many people had agreed they actually wanted this particular software, even for free, and he said zero, so far. But that didn't worry him.

---

expensive to use advertisement at all; this might partly be a reflection of their lack of sophistication in paid-advertising, but also the ROI of AdWords has continued to worsen.

Your knowledge of one slice of a market doesn't automatically set you up for other slices. You're not starting from scratch, but you have to have an attitude of re-learning, of questioning everything.

Of course if you can *fuse* your special knowledge with an *open mind*, that could very well be an unbeatable combination.

But you have to set your ego aside and actively force yourself to **explore anew with a "child-like" mind**. Use that Rolodex to set up meetings and sales calls, but don't assume you know what they're going to say (p. 239). Use that experience to come up with plausible theories, not to make decisions.

It won't happen unless you force yourself to do it.

After all, you're the expert.

Chapter 132:
Being who you are, while becoming better

Whatever your position on meat dresses, no one today is more vocal about being true to yourself than Lady Gaga:

> God makes no mistakes.
> I'm on the right track.
> I was born this way.

As she's said in interviews, she's not speaking only to the LGBT[1261] community; she's encouraging everyone who is seeking their identity and still looking for permission to be that person in public.

It's so simple and trite: Be yourself.

The benefits are obvious: Fulfillment, happiness, even success. Surely you're more likely to be successful at any venture so long as your natural excitement leads to merry obsession, which leads to hard work and long hours, applied to a field aligned with your innate talents.* Even in the worst case—complete failure—it was fun and worthwhile. Years from now you'll look back and say, "I lived a good life, and it was *my* life."

So why is it so **hard?**

Not just hard to live true to yourself, but surprisingly difficult to even know what "yourself" is?

And further, **what about self-improvement?** It's sensible to embrace your faults and work around them (p. 543), but what about tackling them headlong? If you're afraid of public speaking, is it smart to pick a career that ensures you'll never grace a stage, or is it wiser to throw yourself into a stand-up comedy night-class that could exorcise your demon?

I used to think that in all things I had to be the best. I was upset if I didn't get the blue ribbon in the piano recital, the gold medal on the music theory test, and the 1st-place trophy in the Karate tournament.

In high school, however, I started internalizing that there will always be people smarter and better than me at anything, so the new challenge was to (1) be the best at *one thing* (telling a computer what to do), and (2) be constantly improving in other things. So I didn't need to beat Alex Saltman on the Math Team, but I worked to beat my own scores in national competitions year over year. I didn't have to outpace the endless stream of Korean piano prodigies but I did play progressively more difficult pieces, and played them as well as I could.

But time erodes your sharpest edges, and responsibilities accumulate, and you realize that even just "improving" can be too much work. Today I no longer need to improve my 5k time, I just make sure to run a few 5k's a week, never timed. I don't need to play harder piano

* While this is an essential ingredient—or at least ought to be treated as such—there are many other things which also have to be true (p. 9) for a venture to succeed.

pieces, I just learn something new now and again, optimizing for enjoyment rather than progression.

That's fine for hobbies, but **what about your career or your startup?** You can't just say "Yeah I suck at selling stuff, and I don't care!"

In my case, I don't suck at selling (p. 737), but I constantly struggle with procrastination. Everyone does it to some degree—there's something you don't want to do, so you invent reasons to fulfill that desire. The reasons are all seemingly-logical but actually-bullshit:

- I don't have time for X right now.
  (Time always exists, you're deciding to do something else with that time.)
- It will be more efficient to do X when I'm also doing Y and Z.
  (But someone's waiting for X, and you've also been delaying Y and Z.)
- I'm not in the mood for X, and it will go twice as fast when I'm in the mood.
  (But it's something you don't like which means you'll never be in the mood—or—you haven't been in the mood for weeks.)
- I forgot because I don't have a good system.
  (But there are 100 organization systems and tools, and blogs and books and videos for training.)
- I have better things to do with my time.
  (But then it should be delegated (p. 981) or deleted instead of sitting in your to-do list.)

Still, knowing all this, I persist. It *is* on the to-do list and it *is* a "Next Action," but I deftly roll the deadline over to "tomorrow" and it vanishes for another 24 hours. Congratulations, me, you just outwitted your obedient to-do software.

Besides, procrastination has its advantages. Really! I wrote a whole post about why procrastination is useful for running a startup.[1262] See? *SEE?!?*

I've done everything to cope with procrastination: I've used GTD[1263] for years (and yes, it works). I've been inbox-zero[1264] for

years (and yes, it changed my life). I've had temporary surges of success and guilt-ridden lapses of ineptitude. I've even tried to decide procrastination was a net-positive, hence that article.

So where does that leave me with regard to Lady Gaga's admonition that I should "be myself?" Should I accept procrastination as a given? Keep writing more articles about how I'm actually wise to embrace it? Or should I continue to fight it, because fighting means I'll get more done in less time with less stress, and it means I'm optimizing for the long-run rather than what's easiest in the moment?

I've decided to continue to the fight, because I know something else about "who I am"—I'm a person who strives.

I'm a person for whom "good enough," isn't. I can tell myself it's not important to run a 5k any faster than 26:06, but I'm going to anyway. I can lecture you about how blogging success is about content and time[1265] rather than looks and plugins,[1266] but I still wrote my own plugin to manage exactly how the "retweet" link behaves in the RSS feed.

That attitude is something I see in most successful entrepreneurs. Adam Carolla[1267] says the same thing; he calls it a "motor"—an internal, unstoppable force causing you to just go, all the time, wake to sleep, for decades.

It's why I couldn't just make software facilitating peer code review, I had to make it a so-called "real company," I had to create the modern theory around it, I had to write a book about it (p. 457), and I had to push that book into 70,000 people's hands.

It's why I couldn't just read blog posts by Joel Spolsky (p. 1553) and Jason Fried,[1268] I had to write my own, and I had to get better and better at writing, and promote myself, for no reason or reward other than pure ego. It's why I couldn't just be interviewed twice on Andrew Warner's Mixergy[1269]—already an honor. I had to push for a third appearance so I could interview Andrew on his own show.[1270]

It's why I couldn't just be retired after the sale of Smart Bear (p. 45), writing blog posts and talking to entrepreneurs. New ideas foisted themselves on me, and I had to run through bad ones until I found a

good one (p. 845), and I had to go start that company (p. 9). It's why so many entrepreneurs are serial entrepreneurs—I like to say "You do the third one for the same reason you did the first one—because you are *compelled* to."

But the motor also **creates problems common to most entrepreneurs**, no matter how old or successful:

"Spread too thin" syndrome.
You're interested in everything, you're good at many things, so you get involved in too much stuff. Now you don't have enough time for any of them; most suffer as a result, probably even worse off than if you weren't involved, because then other people could plan accordingly instead of believing they can rely on you.

"Shiny new thing" syndrome.
You're bored as soon as a project leaves the childhood of "mostly creative" and enters the adolescence of "mostly execution." So you don't give projects the lasting attention they need to succeed.

"Work all the time" syndrome.
This works better when you're young, but even then you'll burn out. I did, and many other famous workaholics did, repeatedly, though they rarely admit it. Andrew Warner is honest enough to admit publicly that he sold his company for less than he should have just because of burnout. I am too.[1271] Pulling 70-hour weeks catches up to you. Period.

"Not good enough" syndrome.
Whether you're actually a perfectionist or just a control-freak, you feel like nothing is ever finished, ever done, ever enough. And when you implicitly believe others will discover your ineptitude, or that others are able to be perfect where you're not, you (like me) have real problems (p. 457).

At the end of the day, these things are all manageable by acknowledging them and acting accordingly. Learn to say "no" to new projects (p. 621) (allowing you to continue a healthy obsession over a few). Go on a trip without the laptop so you have no choice but to be "unproductive." (Do that once for a week and you'll be so amazed at your

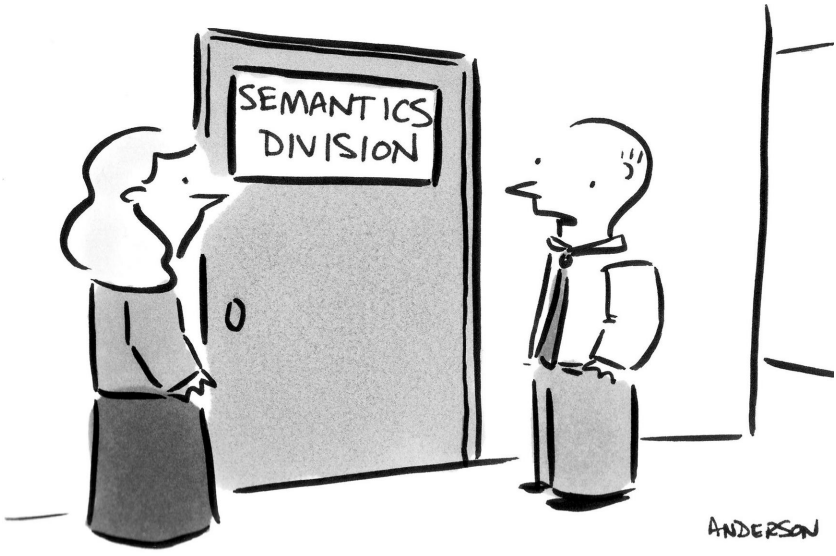
increased productivity, energy, and happiness when you return that you'll never doubt this tool again. Take it from this workaholic.)

So "be yourself," yes. Don't vomit out some stupid marketing prose (p. 627) on your website; decide what you believe (p. 939), then run your company accordingly and publicly[1272] and you *will* be more proud, more fulfilled, and likely more successful. You can't fight every foible, nor should you. You certainly shouldn't compromise your ethics and your sense of taste and awesomeness, because those are some of the few things that differentiate you and your company in the world.

But neither should you stop striving, improving, learning, and developing "who you are."

Feed your motor.

Chapter 133: Specificity: A weapon of mass effectiveness



"We're really more of a department."

credit [273]

My single best advice about writing—whether for marketing copy, blogging, a sales pitch, an investor pitch, or even humor, is:

BE SPECIFIC

All done, you can stop reading now. (As if I have the power to decide when you stop reading.)

Writer's Workshop

Sometimes it's easier to demonstrate than to preach, so let's take a simple statement and see how being specific makes it more powerful, more interesting, and even funny.

Here's our starting point:

> Experts say Twitter usage is increasing, but in many industries your marketing efforts are better spent in other channels.

No generic words

The first and easiest step is to swap generic words for specific ones. By "generic" I mean words that span broad concepts instead of conjuring a specific image. In our example, these include:

- usage
- many
- effort
- better

Generic words are a sure sign of lazy writing. You can't be bothered to describe *what* people are doing on Twitter so you say "usage." You can't quantify the number of industries so you say "many."

Besides being boring and uninspiring, generic words are interpreted differently (p. 997) by different people, so it's not clear what message will be received on the other end of the Internet connection.

Here's a stab at converting generics into specifics; see how much clearer the point becomes:

> Experts say people increasingly base buying decisions on Twitter conversations, but in non-technical industries Twitter penetration is still low, so other marketing channels have a higher return on your time investment.

From truth to accuracy

After disposing with the obviously-useless generic words we're still left with words which, while technically correct, are still not contributing enough to the meaning and persuasiveness of your writing.

Take the word "expert." Yes "experts" talk about Twitter, but what sort of expert (p. 1427)? When I think of Twitter experts I think of Chris Brogan[1274] and Tony Hsieh.[1275] But they're not merely "experts" in the sense that they have a deep knowledge of Twitter, they also actively promote Twitter and teach others how to become experts themselves.

This is an important distinction, because experts in other fields are often not as evangelical; an expert classical guitarist might not, in fact, be a teacher or care about convincing millions of people to pick up a 12-string.

In our workshopped example we're making the point that Twitter experts tend to promote Twitter without qualification; using the word "evangelist" instead of "expert" is more specifically what we mean to say.

Replacing "expert" and a few other words (e.g. "higher return" → "profitable" and "people" → "consumers"), see how much more evocative the statement becomes:

> Twitter evangelists encourage consumers to base buying decisions on Twitter conversations. But Twitter hasn't made inroads in non-technical industries, so traditional marketing channels might still be more profitable.

Concrete examples

If your primary goal is brevity, examples are bloat. Otherwise, examples improve persuasive writing in several ways:

- Examples clarify abstract arguments.
- Examples make arguments more believable.
- Examples make arguments more difficult to counter.
- Examples make it easier for readers to apply your points.
- Examples make it easier for readers to do their own research.

Consider how much more abstract this article would be without the workshopped sentence! Speaking of which, let's update our Twitter statement:

> Evangelists like Chris Brogan[1276] and Mike Volpe[1277] say consumers increasingly base buying decisions on Twitter conversations. But Twitter hasn't made inroads in non-technical industries like agriculture, construction, and retail, so in those cases it's more profitable to stick with traditional marketing channels like direct mail and resellers.

See how much more evocative it is to put a name to the evangelists; even if you've never heard of those guys, knowing their names makes it tangible.

The easiest way to be funny

It's odd but true that almost any statement can be made funny merely by being specific, and there's so many ways to do it.

In the following examples, note how the individual nouns, verbs, and adjectives are specific.

Exaggerate to extreme
Instead of: "Evangelists enjoy pointing out how consumers use Twitter."

Write: "Evangelists wet their pants every time someone uses Twitter to find a coffee shop."

Exaggerate to banality
Instead of: "Evangelists enjoy pointing out how consumers use Twitter."

Write: "Evangelists proudly point to Twitter's popularity, enabling people of every creed, color, persuasion and nationality to join the global conversation on what they had for lunch."

Invent an example
Instead of: "... hasn't made inroads in non-technical industries."

Write: "Farmers looking for a deal on a new tractor aren't peeling iPhones out of their Wranglers and thumbing out tweets through work gloves."

Highlight the absurd thing
Instead of: "Evangelists enjoy pointing out how consumers use Twitter."

Write: "Evangelists tells us we can use Twitter to gather 'advice' culled from the stray comments of millions of strangers and bots."

Blow something out of proportion
Instead of: "Evangelists enjoy pointing out how customers use Twitter."

Write: "Evangelists say your customers are on Twitter, but mostly they're twittering about how to twitter, and about how everyone else is twittering, and most of *those* people are bots, and about how all those human non-twitters are missing out on this incredible exchange of knowledge."

Transfer the concept to another context
This is the principle behind my satirical post on the "rules" of social media.[1278] (In fact that post demonstrates all these techniques!)

Putting it all together

Here's some lovely examples of other writers using the above techniques for humor and effectiveness:

- Specificity and the Art of Being Wrong[1279] (Naomi Dunford)
- Painless Functional Specs—Part 4[1280] (Joel Spolsky)
- Don't Follow your Passion[1281] (Amy Hoy)

Chapter 134:
Startup Exercise: What can't be solved with money?

SOLVED WITH MONEY · NOW THEREFORE…

"Mom & Dad are my primary funding, but I get plenty of soft money from my grandparents."

credit 1282

When pitching an investor, one of the key things you need to communicate is how **the important problems facing your business can be solved with money**, because money is what they are providing.

Even if you're not raising money this is a useful exercise, because if you're good at the things money *can't* buy, you'll remain competitive even when confronted by a well-funded competitor (p. 295).

So, it's useful to separate the aspects of your business that could be improved with money alone, from what instead requires time, attention, intelligence, and even luck (p. 1035).

Team

Finding good people is almost impossible. I have this blog, a nice Twitter following, attention in my local community, and social "favors" I can call in, and it's still almost impossible. And the thing about great people is they always have options: Existing job, other excellent offers, the freedom to not work for a while, etc..

If someone handed me a million dollars it wouldn't help me find an awesome person; they're just scarce.

Once I find someone, however, then money helps a lot, because I could satisfy large salary requirements, handle relocation, provide a signing bonus, lease a car, or whatever else is necessary to make money "not the issue."

So if you're raising money and you don't have your team assembled, the investor knows that handing you $200,000 won't fix that. Whereas if you have the team, but they're working only in their spare time because day jobs pay the mortgages, that's a good reason to raise money.

Marketing

Marketing of most sorts can be solved with money. Money means you can spray ads everywhere (p. 913). Money means you can try 20 campaigns (p. 1027) even if most utterly fail (p. 1369). Money means you can try 50 titles/descriptions/landing pages through AdWords until you find one that converts decently. Money means you can fund a Freemium offering (p. 1385).

One big exception is anything that requires authority (p. 627), like blogging and social media. You can't buy authority. You can't buy that *kind* of attention, where people listen because they want to listen, not because they're interrupted, where they interact with you because they respect your opinion and enjoy your style.

Social media marketers have already beat this point to death. But if your business requires the modern "tell-your-friends" social-media-style punditry, you'll have to do that without money.[1283] If it's just a matter of pouring more marketing dollars in the top so you can crank out more revenue in the bottom, that's a machine investors like to invest in.

And so forth

You see the pattern, so **let's accelerate**; what can money solve?

- Writing more code, faster, yes. Knowing (p. 817) *what* to write, no.
- Acquiring more leads, yes. Know how to convert them, no.
- A prettier website (p. 853), yes. What text to put on the website, no.
- Leeway to make mistakes, yes. Philosophy of learning and using mistakes (p. 1261) to improve, no.
- A QA team to improve quality, yes. Knowing when a bug isn't important enough to prevent shipping, no.

- Time to outlive a recession,[1284] yes. Making something people want to buy (p. 275) even during a recession, no.

The pattern: Concepts, behaviors, knowledge, and process cannot be fabricated with money, and possibly cannot even be accelerated. Once you know what to do or, more likely, you know how to learn quickly, then money becomes an accelerant.

Even more briefly using the Explore vs Execute modalities (p. 521): Execution can be solved with money; exploration cannot.

NOW THEREFORE…

How can you use this principle to raise money?

credit[1285]

"You know, for just a small increase in my allowance,
I could make bedtime a lot easier."

1. Emphasize how spending money will improve specific things which, today, are broken or missing *only* because money is lacking.
2. Show proof of your ability to master the things money cannot buy—your ability to learn, change, and improve.
3. For those things money can accelerate, and for which you've spent a teeny amount of money, show how well you've done with that. It's easy to imagine more money bringing more results.
4. For those things money can't solve and which you haven't mastered, bring it up before the investors do. And have a plan, proving you're self-aware and thorough. If you don't have a clear, plausible plan, don't raise money yet.
5. Specifically, don't raise money if you haven't assembled the core team.

Let me make point #2 tangible.

I cannot count the times I've heard someone proclaim how adaptable they are. "I'm in love with A/B testing." "I'm not afraid to admit when I'm wrong." "I listen to my customers." "I'm a big fan of Eric Ries."

Yeah, you and 60,000 other people (p. 1459). I'm supposed to be impressed?

Since everyone and their dog is now an expert in Lean Startup, you need to *demonstrate*, not regurgitate platitudes. Tell me about how and why your pitch changed when you vetted your idea with potential customers (p. 239). Walk me through a screenshot (p. 1517) of your app, explaining the customer feedback that lead to each part. Show me your company dashboard with the numbers that matter (p. 645), explaining why each number is there, what you expect those numbers to do, and what you do when they don't go as expected. Tell me about your first idea which turned out to be wrong but lead you to the second (p. 1575), and how much easier it was to sell the second. Tell me what happened when you tried different price points (p. 515), and why you settled on the current one. Show me which part of your landing pages improved with A/B testing (p. 913) and which required

old-fashioned creative thought. Tell me in your customers' own words why they're willing to pay (p. 1469) you for any of this.

When you demonstrate that you can do these things, what you're really saying is that you know how to think, how to overcome roadblocks, how to figure out what to do.

All things money can't buy, but exactly the things which, when combined with money, make companies most likely to succeed (p. 379).

If you're not raising money, you still need these skills, exactly because you don't have the money to waste!

Things money cannot buy are still the most valuable things.

Chapter 135:
Reputation isn't as powerful as you imagine

"The fame is fine, and the money's great, but sometimes I miss the plumbing."

credit 1286

The most common reaction to my recent announcement* of starting a new WordPress hosting company[1287] was that this blog provides me with a ridiculous unfair advantage.

But did it?

Here's what people said:

> "You're starting with 18,000 prospects. **How convenient.** ;) I wish I had that kind of mailing list starting out."
>
> "**You're doing this the easy way**, publishing this post so that thousands of users see it."
>
> "It is, of course, simple to talk about how easy it is to be popular, when you're the already established prom queen."

Fair point, but what *actually* happened after that post announcement? How unfair was this advantage?

Interestingly, Eric Sink[1288] got the same reception years ago when he launched[1289] a little company of his own. It's worth hearing Eric defend himself because it's just like my scenario, but because this happened six years ago I can reveal his results at the end of this post:

> "Reactions to my Winnable Solitaire experiment were mostly positive, but several people claimed my experiment was "unfair" or "invalid". In a nutshell, they argued that because I am already "famous" for my writings about the business of software, I have an advantage that is not available to my readers. My experiment is therefore meaningless because I did not duplicate the conditions a regular person would be facing when trying to launch their own micro-ISV."

Let's start with the results of my announcement:

---

* Editor's Note: This was written in 2010.

Damon Chen
@damengchen

💯 agree! Let me show you my proof 👇

I have 60k followers here. Launched quickyAI.com (a software) on Product Hunt yesterday.

The result? $0 in sales.

credit [1291]

WPEngine got two new signups. Only two. That with 18,000 wonderful, loyal, friendly, supportive RSS subscribers and as many page-hits from Twitter and HackerNews.

Not exactly the massive boost you *nor* I was expecting. I figured on 10-20 new customers at minimum and dreamed of 50. I was wrong by an order of magnitude.*

Eric had a similar result: One month into his Winnable Solitaire experiment he had sold a total of six copies.[1290] Hooray for fame.

Others too.

Let's put this into broader context: At WPEngine we had 50 *paying customers* (not prospects) before my post went live. Most are paying $49/mo, a few pay north of $1000/mo (large blogs with serious traffic). So whatever we did *without* the advantage of this blog was far more important, at least for getting initial customers. *(I'll explain exactly how we did that in future posts.**)*

Still, the blog *was* instrumental in getting those first 50 customers, but not because I'm able to push WordPress hosting onto 18,000 unsuspecting victims, a.k.a. 1000 "true fans." The blog did help with building the team.

---

* Another example of why you cannot predict the future (p. 193).

** Editor's note: It took 13 years, but I finally wrote up that process for customer discovery (p. 239), as well as a complete roadmap for Product / Market Fit (p. 9).

Tom Bilyeu ✓
@TomBilyeu

I have 4M followers who won't buy my products.
Everyone says build an audience first.
Then sell them anything...

I did that.

Built a massive mindset audience.
Got hundreds of millions of views.
Launched mindset comic books.

Failed.

Built a YouTube following.
Tried to sell them games.

Failed again.

Tried telling my audience "but it's about mindset!"
They didn't care.

credit [1292]

Here's the truth:
Big audience ≠ Big sales

It's no secret that the team is a critical factor in a startup's success; have you ever heard otherwise? But there's precious little advice about finding and gathering that stellar team. Interviews on the subject invariably turn up explanations like "we went to school together" or "we worked together" or "we met at StartupWeekend[1296] " In short, you put yourself in an environment where you're likely to interact with other intelligent, capable people, and hope that you find someone socially compatible who is also crazy enough to want to do a startup. It's a good strategy, and anyway what else can you do?

Rob Walling ✓
@robwalling

I've had to say this a LOT over the years, and it feels like every couple months I revisit the topic on @startupspod.

I've invested in more than 170 SaaS companies, many of them wildly successful. Less than 5% of them have any kind of notable audience.

credit [1293]

Laura Roeder
@lkr

Should you build an audience before you launch your #SaaS?

Absolutely freakin not.

Here's why:

(Here is her excellent article [1295] answering why.)

credit [1294]

I knew I needed a killer team for WPEngine—not just "capable," but a group that **would itself be an unfair advantage**. See, "WordPress Hosting" is already a commodity, with every hosting company on Earth offering something at every price point (p. 515) from $0/mo, $5/mo, $15/mo, $40/mo, and even $500/mo + $200/hr consulting fees. **In a mature market you need severe points of differentiation** or moats (p. 761), and one of those (I felt) had to be the team itself.

We needed someone like Aaron Brazell.[1297] Aaron is a WordPress core contributor and the author of WordPress Bible [1298] (Wiley). He's famous enough that strangers at WordPress conventions ask for auto-

graphs of their dog-eared copy of his infamous tome. He has seventeen zillion Twitter and blog followers, most of whom are themselves active in the WordPress community. He knows all the major players in the industry including the key folks at WordPress.com, BZ Media (the CopyBlogger media group), ProBlogger, and members of the press at Mashable, TechCrunch, and others.

Maybe with an Aaron we'd have a chance. His network should provide an ocean of free leads. His reputation transferred to the company would bless us with instant credibility. His press connections should give us pops of traffic and external legitimacy. His intimate knowledge of WordPress internals and roadmap should mean our service is technically superior. That's a lot of advantages! Maybe enough to make or break a little new upstart.

Well **we got Aaron, and it's because of this blog.** When I called Aaron he was charging an obscene (and well-deserved) hourly rate for WordPress consulting in Washington DC, but he was yearning for the startup life. He was ready for the trade-off of definitely less money now in exchange for possibly more money later, and for building something of lasting value instead of the impermanent drudgery of un-screwing hacked WordPress installations.

And the blog sealed the deal. Aaron could have joined (or started-up) any number of startups, but he liked WPEngine because he wanted to do a startup with me. And he wanted to do a startup with me because the blog revealed my attitude, perspective, and credentials.

Aaron picked up, moved to Austin, and has already been instrumental to our success thus far.

So fame does help in important ways—enough even to deserve the title of "unfair advantage (p. 1475)"—but startups are still hard and unlikely to succeed no matter who's at the helm. Case in point? Eric Sink.

Eric's experiment eventually failed. Well, "fail" is a too harsh a word (p. 1261), it's just the one in-vogue nowadays, especially when describing an wonderful experience in which you had fun, learned a lot, grew as a person, and wouldn't trade it for anything, particularly

not a dull, predictable day job. You know, the kind of "fail" that characterizes a lot of software startups.

On sales of $216, Eric sold Winnable Solitaire[1299] for a small sum. Of course neither the exit nor the to-date revenue amounts to anything that anyone would declare a success.

WPEngine's revenue to date is several orders of magnitude more, so hopefully we'll avoid that fate.* Still, our expenses are also orders of magnitude more than Eric's, and as I hope I've shown, although we have decided advantages, it's never an easy road.

But then, if it were easy it wouldn't be worthwhile (p. 737), right?

* Editor's Note: We went on to become a Unicorn, and in 2023 are still growing and profitable.

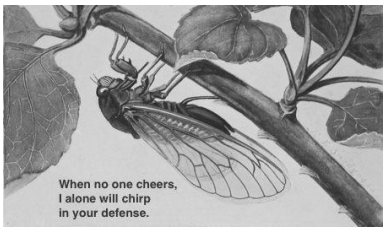
Chapter 136:
If you build it, they won't come, unless...

HONESTY · INFECTION · CRYING · ADS
CELEBRITY

Ask a technical founder about her startup, and she'll proudly describe her stunning software—simple, compelling, useful, fun. Then she'll describe her cutting-edge platform—cloud-based, scalable, distributed version control, continuous integration, one-click-deploy. Maybe you'll even get a wobbly demo.

"Great," I always exclaim, sharing the thrill of modern software development, "so how will people find out about this brilliant website?"

Cue sound of cicadas buzzing.



(Or "crickets chirping" but in Texas the cicadas[1300] are louder.)

Four uncomfortable seconds later, a smile breaks across the founder's face. "Here it comes," I think, "there *is* a strategy after all!"

Except the "strategy" is a tirade of drivel I've heard so many times I can lip-sync as the words spew out the founder's mouth:

- "We're going to A/B-test AdWords campaigns until we discover our hook."
- "We're going to A/B-test our landing pages until the right message appears."
- "We're better than everyone else at SEO."
- "A friend of mine knows how to get popular on Twitter."
- "We're going to get reviews on blogs."
- "We're going to start with our own network and grow it from there."
- "We're going to use an affiliate program so our customers sell it for us."
- "We're putting a 'Retweet' button inside the product to encourage viral growth."

The obvious problem is that **every new startup on Earth says exactly these things. Nowadays the "strategy" above sounds the same as:**

- "We'll have a website so people can read about us."
- "We'll have an email address so people can communicate with us without picking up the phone."

Yes, you're going to do those things, but **since millions of other people are doing that too, you're still invisible.** Visibility-fail. Anyone-gives-a-crap-fail.

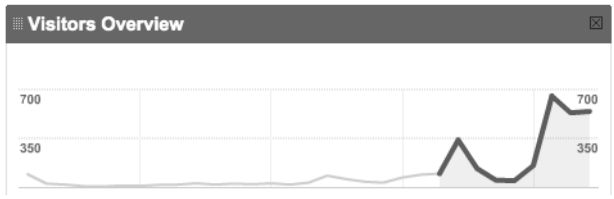
OK, so what can you do to rise above the cacophony[1301] that is the Internet?

FRIGHTENING HONESTY

Balsamiq Mockups[1302] is a ludicrously popular wire-framing tool. The software is good—don't get me wrong—but what sets Peldi (the founder) apart isn't prescient feature selection or bug-free releases, it's his startling transparency. He published revenue[1303] figures even when they were still pathetic,[1304] he pledged loudly and eagerly to give away lots of free copies[1305] to non-profits, and he revealed all his (remarkably effective) marketing strategies[1306] even though it meant competitors would learn them too.

He didn't just have an "authentic voice," he made public promises. That's compelling.

He didn't just "tell it like it is," he gave up his marketing secrets and opened his company books. That's newsworthy.



credit[1307]

Balsamiq's public not-rocket-ship website growth

credit[1308]

Balsamiq: Still a small team of real, caring people, who don't dress up for photos.

This isn't merely "being human" and all that claptrap, it's almost *too much honesty*, like when you ask someone how it's going and they tell you about a weird pustule on their middle toe that's been oozing since last Wednesday.

In a world where everyone and their brother is "joining the conversation", you have to *truly* bare your soul if you want to compete on the transparency front. It's not for everyone, and I'm not suggesting it ought to be, but if you're going to employ it, don't half-ass it.

INFECTION BUILT-IN, NOT BOLT-ON

Calendly[1309] lets people schedule meetings with you in currently-available time-slots without you having to share your calendar (Figure 1).

The key point is that Calendly's customer ("Reggie" in this screenshot) sends the link to other people, perhaps many times per month. Then, those other people have to *use the product* to schedule a meeting.

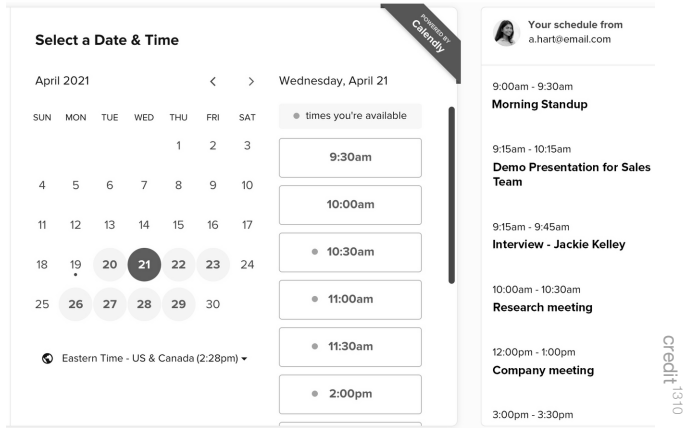


Figure 1

This is the viral step: Having now used the tool, the stranger might use it herself, and so on.

Viral infection works so well, smaller bootstrapped companies like YouCanBookMe[1311] and SavvyCal[1312] compete with Calendly, despite them being large, well-funded, profitable, and growing.

Note that at no point did I say "a button lets people 'like' this on Facebook." I know of no companies who have "gone viral"[1313] because of buttons. Buttons are good—why *not* use them?—but they don't make your product *intrinsically* viral like Calendly.

Which is OK—not all products need to be viral! But if it's not viral you still need a killer method of finding customers, and if it *is* supposed to be viral it better be encoded in the DNA of the application, not bolted on as an afterthought.

MAKING OPRAH CRY

The number one mistake founders make when trying to generate press is talking about what the company *does* rather than telling a compelling *story*.

Does Twitter get press when it helps Iranians fight an illegitimate government or when it creates a new internal IT process to increase up-time? Does Apple win the hearts (yes, *hearts*) of millions because of their obsession with design or because of their development APIs? Does 37signals have over three million users because their software is "better" than the competition, or is it because they motivate designers and entrepreneurs through their writing and philosophy?

Without a powerful narrative, your chances of getting big press and enthusiastic users who spread the word for you approach zero as a limit.

It took me *years* to figure this out at Smart Bear. At first when someone asked what the Smart Bear tool suite was, I would say:[1314]

> Smart Bear makes data-mining tools for version control systems.

It's a description so esoteric that, although accurate, not even a hardcore geek would have any idea what it is, much less why it's useful.

Years later, when we refocused on code review software (eliminating our other 5 products), I got better at describing it:

> You know how Word has "track changes" where you can make modifications and comments and show them to someone else? We do that for software developers, integrating with their tools and workflows.

Better, yes, and for a while I thought I nailed it, but still no press. Eventually (thanks to helpful journalists) I realized that I was still just describing *what it is* rather than *why anyone ought to care*. I left it up to the reader to figure out why she should be excited.

Eventually I developed stories like the following, each tuned to a certain category of listener. Here's the one for the journalists:

> It's always fun to tell a journalist like you that we enable software developers to review each other's code because your reaction is always: "Wait a minute, you're seriously telling me they don't do this already?" The idea of editing and review is so embedded in your industry you can't imagine life without it, *and you're right!* You know better than anyone how another person finds problems you can't see for yourself.
>
> Of course this is true for everyone, in all kinds of work, but developers traditionally work in isolation, mainly because there's a dearth of tools which help teams bridge the social gap of an ocean, integrate with incumbent tools, and are lightweight enough to still be fun and relevant.
>
> That's what we do: Bring the benefits of peer review to software development.

Now the reason for excitement is clear: We're transforming how software is created, applying the age-old techniques of peer review to

an industry that needs it but where it's traditionally too hard to do. *That's* a story.

It took me *five years* to figure out (a) I needed a story and (b) what the story was. It's hard. But one story beats a pile of AdWords A/B tests (p. 913).

ADVERTISING → [TRANSMOGRIFICATION] → REVENUE

Yeah yeah, nowadays marketing is about "relationships" and "authenticity (p. 627)" and other things which cost time but not money (p. 1445). It's all I hear about anymore.

But don't be so quick to jettison the idea of spending money to make money. Advertising isn't dead; you can still buy eyeballs. I'm not talking about arbitrage strategies like buying AdWords that link to a page of ads, but most companies on Earth don't depend on "joining the conversation" to acquire customers.

It *sounds* simple: The average cost of acquiring a customer is `$C` (advertising, sales, support, doing demos) and the lifetime revenue you get from that customer is `$R`, so if `C < R` you have a business. `C` can be driven down with cheaper ads, better lead quality, a more efficient conversion rate, and straightforward trials with minimal tech support.

Of course it's *not* that simple, and many business plans I've seen (unintentionally) omit many of the true costs of acquisition. Read this great interview with Sean Ellis at VentureHacks for a great discussion of how to seek a repeatable, profitable model[1315] where `C < R`, and *then* optimize and grow.[1316] It's a little heavy on the "huge VC-style company" strategy, but you'll come away with a strong perspective on

how to build a machine that turns advertising dollars into (a greater number of) revenue dollars.

CELEBRITY CHAMPIONSHIP

I already beat you to death about how celebrity endorsement can serve as an untouchable competitive advantage (p. 1475), and it's also an answer to how to burst out of the dull roar of Internet marketing.

Take me. I'm no Seth Godin, but consider what I could do if I were a co-founder in a new software development tool company:

- I have personal relationships with the CEOs and other influencers at hundreds of software development companies. During ideation, they would brainstorm. During beta-testing they would be guinea pigs. After release of v3.0 some would be ready to become paying customers.
- I have relationships with editors of nearly all software development publications (on-line and off); I've already published articles with them. Some would help vet our stories, some would publish our articles.
- I've bought ads in every major (and quite a few minor) software development websites, magazines, newsletters, conferences, and webinars. So when it's time to advertise, we'll come in with the right message for the audience and probably cut a deal.
- If you read this blog you're probably a software developer, so even just a few mentions here might be more powerful than $10,000 in A/B tested Google AdWords.
- If we were trying to raise money, my previous success would not only get us the initial meetings but would be a significant bump in our chance of raising it.

While everyone else is mucking about with a new blog, blasting their LinkedIn network with pleading emails, and paying out the nose to test AdWord variants, we're years ahead in the marketing war.

Chapter 137:
Yes, but who said they'd actually BUY the damn thing?

Of hundreds of startup pitches at Capital Factory,[1318] *almost none* had unearthed 10 people willing to say, "If you build this product, I'll give you $X."

Meditate on this: Hundreds of people ready to quit their day jobs, burn up savings, risk personal reputation, toil 70 hours per week, absorb as much stress as having a baby (believe me, I've done both).... all without identifying even ten measly people actually willing to pay for what they're peddling.

Short-sighted, no?

credit[1317]

Editor's note: This article was written in 2010, three years before *The Mom Test* was published.

If you can't find ten people who say they'll buy it, your company is bullshit.

Aren't you sick of every startup blogger on Earth badgering you about this? Steve Blank says "get outside the building," Eric Ries says "seek validated learning (p. 1425)," Sean Ellis says "seek product/market fit (p. 335)," Drew Houston says "the only way to learn on a $0 budget is to talk to people."

I say "find ten people who say they'll buy." I say "get off your ass and produce hard evidence that customers are in your future light cone."[1319] I even tell you how to interview customers (p. 239) and how to do everything else (p. 9).

But you're still not listening. You repeat these mantras at Lean Startup Meetings *but you're not doing it.*

You're understandably scared of been proved wrong, especially now that you're all worked up about the new business idea, and extra especially after you've already told friends and family you're doing this and they're expecting you to complete your quest.

But jeez people, you're not even trying. And worse, **you're inventing lame excuses for why you're not trying.**

One excuse is: "I don't know how to get 10 people to talk to me, before I have a product." OK, here are a huge number of ways (p. 683) that other entrepreneurs have successfully used. Pick one or two.

Full power to forward shields y'all, I'm coming for you.

"I'm scratching my own itch. Since I'm my own target customer, I already know what to build."

Oh! I didn't realize your typical customer is observant enough to recognize monetizable pain, creative enough to invent products, able to convince others to work for free and invest money and time with you, and passionate enough to quit her job to pursue unproven ideas.

Fooey! **By definition, if you're a startup founder you're explicitly *not* your customer.**

"Scratching your own itch" is how all three of my companies started, **but it's only that—the start.** It's the spark of inspiration, not the strategy. It's the grain of sand tickling the oyster, not the pearl.

Look! Smart people agree:

> "Be a user of your own product. Make it better based on your own desires. But don't trick yourself into thinking you are your user." *—Evan Williams*[1320]*, founder Blogger & Twitter*

> *"If the VP of Engineering thinks the target customer is just like him/her, you're doomed. If the VP of Marketing thinks the target customer is just like him/her, you're doomed." —Cranky Product Manager*[1321]

> "Our customers did a lot of stuff that I would never do. We think differently. We solve our problems differently. We have different needs and wants. Repeat after me: You are not your customer." *—Eric Ries*[1322]*, Lean Startup leader (repeating a conversation with a startup founder)*

In fact I challenge you to find *one* founder of a real business who thinks "I'm the customer" is the only market validation you need.[1323] If you still don't believe me, here's a whole article (p. 533) debunking this idea in more detail.

"There are millions of potential customers, so it doesn't matter what only ten of them think. I need to just start; later I can survey and learn something statistically significant."

If there are millions (p. 71), it's trivial to find ten. If you can't find even ten, then either there's not millions or those millions aren't interested in you.

Businesses don't start with millions of customers, they start with one, then ten, then a hundred, and then a thousand. **But most don't get past ten.**

If you haven't gotten ten to at least *say* they'll buy, where do you get your hubris to proclaim that thousands actually *will* buy?

"My customers can't understand mock-ups. I have to build it first."

You shouldn't need screenshots or PowerPoints to convince someone in your target market (p. 317) that what you're doing is compelling. If your concept is so esoteric that you can't describe it in 30 seconds at a cocktail party, it's either too complex or you don't understand it yourself.

Take me and WP Engine. I got *thirty* people to tell me they'd pay $50/mo for this service before I had a company name, a website, a product, a co-founder, or an employee. And don't say it's easy for me because I've done this before—the full story (p. 845) is that I had other ideas which proved to be crap.

Even if I concede that sometimes you need a mock-up, and that some folks can't grok mock-ups, remember that your first customers will by definition be early-adopters who are OK with alpha software.[1324] If you can't find a few of those and get them excited about your product, maybe your product isn't exciting.

"I suck at sales/marketing; I need to build a product so compelling it sells itself."

The world is filled with decent products that make no money. You know this!

Oh fine, you want empirical evidence? Here's a list[1325] of the top 100 Twitter clients, and here's some more.[1326] Now:

- How many do you suppose are decent pieces of software that basically work? (My guess: 80%)
- How many do you suppose produce *any* revenue? (My guess: 5%)
- How many do you suppose produce enough revenue that, after hosting and marketing expenses, they result in a profitable company where the owner doesn't need a day job? (My guess: <1%)

Conclusion: If your goal is a business (not a hobby), **building charming, novel software isn't enough.**

You and I know you have the ability to build cool new software. We agree that will be fun and exciting. But *that's not going to create a business*.

Writing code is what you love, so you myopically decide that's what you'll do (p. 1541). But what you *should* do is just the opposite: Attack the part of the business you're *least* sure of, you're *least* qualified for.

If you're still not convinced, **think of it as project risk management**. In a big software project do you tackle the high-risk, ill-defined stuff first, or do you postpone that to the end? Obviously you address the uncertain stuff first—most of project risk is created by the unknown, so the earlier you can sort out uncertainty the more time you have to deal with the consequences.

I'm making the same argument, except the "high-risk unknown" is "everything that's not code." Your code will be good enough; it's the other stuff that will probably sink your ship—unable to find customers or unable to convince the target audience they should open their wallets.

No sense in postponing it.

"My friend / brother / co-worker / dentist thinks it's a great idea."

Your mother thinks you're smart and good-looking, but that doesn't mean I do.

It doesn't matter what non-entrepreneurs think because they're not versed product/market fit (p. 335), squeezing blood from evanescent budgets, and using Facebook for advertising instead of sharing the latest FailBlog[1327] movie.

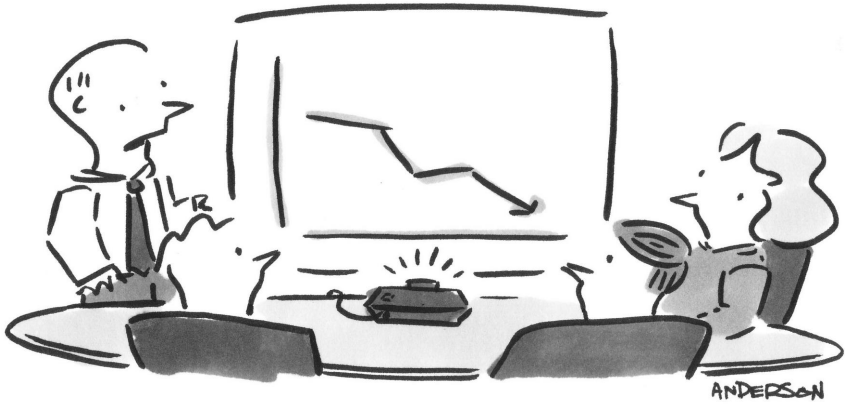
In fact it only barely matters what real entrepreneurs think either, because they're not expert in your problem domain, they might have outdated notions, they might be biased (p. 449) against certain ideas and technology, and they carry baggage from good and bad experiences (due as much to timing and luck (p. 1035) as anything else).

The only thing that matters is that people are willing to give you money!

Business "experts" (p. 1427) can argue all day long that it makes no sense to buy shoes over the Internet, but as long as people give Zappos $1 billion per year, it doesn't matter what experts say.

When ten people say they'll give you money, that's the only validation that counts.

Chapter 138:
Real Unfair Advantages



"I do have a solution, but it means moving to Santa's naughty list."

credit 1328

What if someone copies your awesome business idea?

The question you know you're going to get:

> When I meet an angel investor, he may ask: "What if a big company copies your idea and develops the same website as yours after your website goes public?"
>
> How can I answer this question?

No, the question is: What are you doing *now* knowing that a big company *will* copy your idea?

No, wait, the real question is: What are you going to do when another smart, scrappy startup copies it, and gets $10m in funding, and is thrice featured on TechCrunch?

No, wait, I'm sorry, the *real* question is: What are you going to do when there are four totally free, open-source competitors?

No wait, I forgot, actually the question is: What happens when employee #2 makes off with your code and roadmap and marketing data and customer list, moves to Bolivia, and starts selling your stuff world-wide at one-tenth the price?

The good news: **There *are* good answers to these questions!**

The bad news: Almost no one I talk to has good answers, *but they think they do*. And that's fatal, because it means they're not working towards remedying that situation. Which means when one of the above scenarios happens, it will be too late.

The first step is admitting you have a problem.

Anything that *can* be copied *will* be copied, including features, marketing copy, and pricing. Anything you read on popular blogs is also

read by everyone else. You don't have an "edge" just because you're passionate, hard-working, or "lean."

The only real competitive advantage is that which cannot be copied and cannot be bought.

Like what?

Insider information

They say the only way to consistently make money on Wall Street is to have insider information. Unfortunately it's not a joke, and although it's illegal (and people occasionally go to jail for it), those in the know will tell you it's the norm.

Fortunately, using intimate knowledge of an industry and the specific pain points within an industry is a perfectly legal unfair advantage for a startup.

"What happens to the stock when I do *this*?"

credit 1329

Here's a real-world example of how this advantage manifests. Adriana has been a psychiatrist for 10 years; she understands the ins and outs of that business. During a lull in her practice she got a serendipitous opportunity to shift gears completely and ended up leading software product development teams. (Turns out that for big-business project management it's more valuable to be a sensible thinker and counselor than to be an expert in debugging legacy C++ code.)

Now Adriana has an epiphany: Traditional practice-management software for psychiatrists totally sucks; she knows both the pain points and the existing software first-hand. But now she has the vision and ability to design her own software, capitalizing on modern trends (e.g. a web application instead of cumbersome installed applications) and new interpretations of HIPPA regulation (which allows web-based applications to store medical records like patient histories).

Adriana holds a unique position: Expert in the industry, able to "geek out" with her target customer, yet capable of leading a product team. Even if someone else saw Adriana's product after the fact, it's almost impossible to find a person—or even assemble a team—who has more integrated knowledge. At best, they could copy. Of course by then Adriana has moved on to version two.

Single-minded, uncompromising obsession with One Thing

A popular comment on the previous post[1330] was that a "Unique Feature" *could* be a competitive advantage in some circumstances. Some examples of a feature being a company's primary advantage are:

- Apple compromises everything in the name of design. Their products are over-priced (magically being profitable at half the price 12 months after release), buggy (how many iOS debacles have there been?), and every experience I've had with their tech support has been atrocious, but man their stuff looks and feels nice! (I'm typing this on an Air and there's an iPhone in my pocket, so no Apple fan-boy mail please.)
- Google's search algorithm was just better, therefore they won the eyeballs, therefore they were able to monetize. Sure Bing and Yahoo are good *now*, but the advantage lasted long enough.
- Photodex is a little company you've never heard of I worked for in Austin in the 90s. We made an image browser with thumbnail previews so you didn't have to open each file individually to see what it was. (In the 90s, y'all, before that was built into all the operating systems!) Our advantage was speed. Not the best, not the most stable, didn't read the most formats, didn't have the most features, just "fastest." For many users of that product, speed wins; Photodex now makes tens of millions of dollars a year, and "speed" is still the only point on which they will not compromise.

However it's not enough for a feature to merely be unique (like my mini-browser (p. 1517)) because it's still easily duplicated. Indeed, most of the innovations we've made at Smart Bear[1331] in the art of code review have already been duplicated by both commercial and open-source competitors.

Rather, this requires ***unwavering*** **devotion to the One Thing that is (a) hard, and (b) you refuse to lose, no matter what.**

Google has spent hundreds of millions of dollars on their search algorithm, the single biggest focus of the company even today, a decade after they decided that was their One Thing. They refuse to be beaten by competitors or black-hat hackers, whatever it takes.

37signals can build simple—almost trivial—software and earn three million customers because they absolutely will not compromise on their philosophy of simplicity, transparency, and owning their own

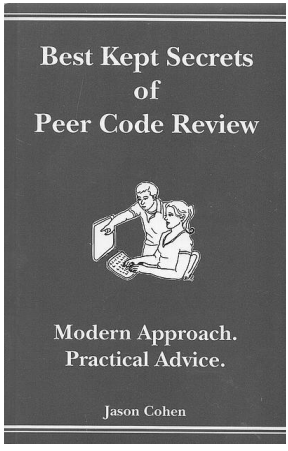
company, and that's something millions of people respect and support. Competitors could build trivial web applications too (as Joel Spolsky[1332] is fond of saying, "Their software is just a bunch of text fields!"), but without the single-minded obsession it's just software with no features.

To remain un-copyable, your One Thing needs to be not just central to your existence, but also difficult to achieve. Google's algorithm, combined with the hardware and software to implement a search of trillions of websites in 0.2 seconds, is hard to replicate; it took hundreds (thousands?) of really smart people at Microsoft and Yahoo years to catch up. 37signals' ranting platform—a blog[1333] with 131k followers and a best-selling book[1334]—is nearly impossible to build even with a full-time army of insightful writers.

"Being hard to do" is still a true advantage, particularly when you devote your primary energy to it.

P.S. For more, here are detailed examples[1335] of how this mindset also sets up your sales pitch.

Personal authority



Chris Brogan commands $22,000 for a single day of consulting in an industry (social media marketing) where all the information you need is already online and free. Joel Spolsky makes millions of dollars off bug tracking—an industry with hundreds of competitors and little innovation. My company Smart Bear sells the most expensive tool of its kind. How did we earn this powerful authority, and how can you earn this overwhelming advantage?

I'm a great example of someone who wasn't an authority on anything, but built

that authority over time to the point where now my company (Smart Bear) is untouchable as the leader in both revenue and ideas in the area of peer code review.

Not only was I not an expert on code review prior to building a code review tool, I wasn't even an expert on software development processes generally! I didn't give lectures, I didn't have a blog, I didn't have a column in Dr. Dobbs magazine, and most interesting of all, I didn't even know "code review" was going to be what made the company successful (p. 1575)!

Unfortunately all this "authority" crap takes years of expensive effort,[1336] and even then success is probably due as much to luck[1337] as anything else, so is it worthwhile? Yes, exactly *because* it takes years of effort and a little luck.

Authority cannot be purchased. You can't raise VC money and then "have authority" in a year. A big company cannot just decide they want to be the thought-leaders in their field. Even a pack of hyper-intelligent geeks cannot automatically become authorities because it's not about how well you can code.

But how does authority convert to revenue? Here's one tiny example:

> I give talks on peer code review at conferences.[1338] My competition pays thousands of dollars for a booth, then spends thousands advertising to attendees begging them to come to that booth, then gives sales pitches at the booth to uninterested passersby who are also being bombarded by other pitches and distracted by the general hubbub.
>
> Whereas, because I'm a known authority on code review and software development, I get to talk for an *entire hour* to a captive group of 100 people, self-selected as interested in code review. After the talk typically 5-20 people want to chat one-on-one. Some head straight to the booth to get a demo; for many I give a private demo of the product on sofas in the hall-

> way. **It's not unusual to get $10,000-$50,000 in sales** over the next three months from people who saw me at that talk.

That's just one example! Now add to that: What's the effect of a blog that tens of thousands of people read? What's the effect on sales of my writing the book that's the modern authority of code review?[1339]

Authority is expensive and time-consuming to earn, no doubt. But it's also an overwhelming, untouchable competitive advantage.

(P.S. I'm hoping that the authority I'm slowly earning from this blog will help when I launch my next venture. That's not *why* I blog, but I certainly will leverage it when the time comes!)

(P.P.S. I apologize for blatantly abusing the word "authority," considering I just lambasted everyone who does (p. 627) things like that.)

The Dream Team

The tech startup world is littered with famous killer teams: Gates & Allen, Steve & Steve, Page & Brin, Fried & DHH.

In each case, the founders were super-smart, had complimentary skill sets, worked together well (or well enough to get to important success milestones), and as a team represented a unique, powerful, and (in retrospect) unstoppable force.

Of course that's easy to see in retrospect, and retrospect is a terrible teacher (p. 449), but the principle can work for any startup, especially when your goals are more modest than being the next Google.

Take the success of ITWatchDogs,[1340] the company I helped bootstrap and eventually sell (before Smart Bear). The elements of our Dream Team were obvious from the start:

- **Varied skill sets.** One experienced startup/business/salesman (Gerry), one proven software developer (me), one proven hardware developer (Michael).
- **Common vision.** We agreed what the product ought to be and that the ultimate goal of the company was to sell it.
- **Insider knowledge.** Gerry had done another successful startup in the same space, I had deep experience with the language and tools for embedded software, and Michael had decades of experience building inexpensive circuits and processors.

Of course a Dream Team doesn't guarantee success but it **significantly reduces the risk** of the startup, and furthermore is difficult for the competition to duplicate.

This is especially true when someone on the team is already successful in their field, e.g. with a massively successful blog or a big startup success under their belt or a ridiculous Rolodex. Since those are the kinds of competitive advantages that can't be bought or consistently created, having that person on the team is by proxy a killer advantage.

P.S. This is the primary competitive advantage in a new startup I'm working on right now (to be announced soon), so shortly you'll see another example of this theory and—better yet!—you and I both will witness over the subsequent months whether or not this really resulted in a killer advantage! (Yes of *course* I'll share details!)

(The right) Celebrity endorsement

Hiten Shah's[1341] third company is KISSMetrics.[1342] On the surface, it's yet another "marketing metrics" company. This is a crowded, mature market with *hundreds* of competitors in every combination of large/small, expensive/mid/cheap/free, and product/service/hybrid.

But Hiten has something none of those competitors has: Investors and mentors who are celebrities in exactly the market he's targeting.

Folks like Dave McClure,[1343] Sean Ellis,[1344] and Eric Ries,[1345] all of whom not only help via conference call but actively promote KISSMetrics on their blogs, Twitter, and personal appearances.

How much advertising will it take for competitors to overcome Hiten's endorsements and exposure? Even if a competitor also wanted celebrity endorsement, *these guys are taken*, and in any field there's a limited number of widely-known and respected authorities.

Many competitors have more features than KISSMetrics has. I can see the sales pitch now...

> The customer objects: "Gee it would be nice to have all those features," and Hiten responds "Well not really, because Dave, Sean, and Eric all say that those features are actually distractions and don't add to your bottom line. Our features are the right ones, as evidenced by these 20 companies that have shown increases in revenue."

Just on the basis of these advisors, Hiten will get hundreds if not thousands of customers. You can't buy that kind of jump-start, not even for millions of dollars, because it's not about faceless leads who saw KISSMetrics in an ad, it's people who *trust* Hiten because of his association with other people they already trust.

P.S. If you're raising money, investors love to see a co-founder or even just an advisor who has been successful before. The VC game is more lemming-like than most care to admit.

Existing customers

...or as Frank Rizzo says:[1346] Open your ears, jackass!

Everyone you've ever sold to (and those who trialed but abandoned[1347]) possess the most valuable market research imaginable, and it's the one thing a new competitor absolutely will not have.

This is kind of a cheat, because everyone says "I listen to my customers," which (nowadays) is just as bullshit as "We're passionate," but

it's true that if you're actively learning from your customers *and* you never stop moving, creating, innovating, and learning, that puts you ahead of most companies in the world.

As a company becomes successful it gains momentum, which means that it's going in one direction with one philosophy. Like physical momentum, change becomes harder to affect. It's logical; for example at Smart Bear we have 35,000 users, so making a drastic change to the user interface or typical workflow would mean too much retraining, even if the end result is better.

Even "cool, agile" companies like 37signals are trapped. They've been so clear and confident in their philosophy of "do less," they cannot go after markets where "less" is not more but, actually, just less. For example, with more than a few sales people in a traditional sales organization it's impossible to use Highrise—the folks-of-many-signal believe pipeline reports and geographic domains and integrated campaign management are unnecessary complications, but actually it's Highrise that is unnecessary.

Of course the world *is* changing, and in particular your customers are changing. Normally this leaves room for the next competitor, but if you're already entrenched you can leverage your existing status, insider knowledge, and revenue stream as long as you're willing to change too.

You have more money, you're better known, you have existing happy customers to help spread the word, you have employees to build new things, and you have more experience with what customers actually do and actually need, which means you should have the best insight.

Any new competitor would kill for just one of these advantages. If *you're* not using them, how silly is that?

Zoho made exactly this argument[1348] to explain why they're not terribly worried that Microsoft is now a direct competitor:

> **Companies don't get killed by competition, they usually find creative ways to commit suicide.** Office 2010 will be the

> end of Zoho, if we stop innovating, stop being nimble and flexible in our business model. Then again,if we stop all that, Zoho will die***anyway***, no Office 2010 needed to do the job.

37signals is trapped inside their self-imposed philosophy, but you don't have to be.

Go git 'em

Imitation might be the sincerest form of flattery, but it's still sucks when someone does it to you.

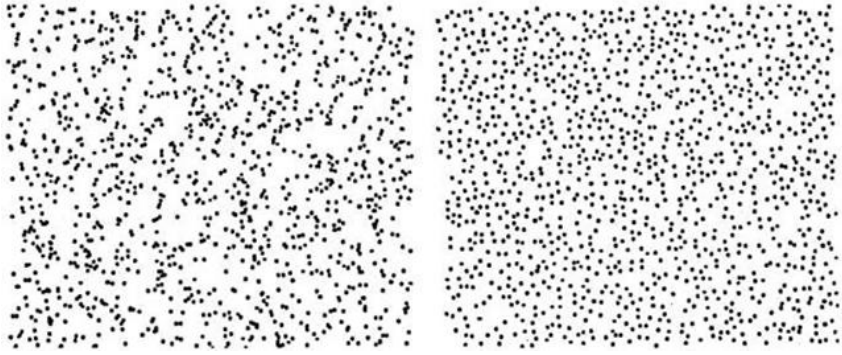
Of course you can still battle it out in the marketplace, but you need *something* that can't be duplicated, something they could never beat you on, then hang your hat on that[1349] and don't look back.

Don't despair if you don't have an unfair advantage *yet*. I didn't either when I started Smart Bear! Almost no one does, at the beginning (p. 429). But I built toward having some, and eventually earned it.

You will too, if you're intentional about it.

Chapter 139:
The Pattern-Seeking Fallacy

PATTERNS IN CHAOS · SEEKING COMBINATIONS
UNFAIR FAIR COINS · MARKETING & SALES
THEORIES



Which of these sets of dots were placed randomly? We think “right,” but the answer is “left.” Humans are bad at seeing the difference between patterns and randomness. (From Steven Pinker’s *The Better Angels of Our Nature*[1350] 2011)

What do these have in common?

- "This pitcher has retired 5 of the last 7 batters."
- "We tried 10 AdWords variants and combination D is the clear winner."
- "The Bible Code predicted the Sept 11 attacks 5,000 years ago."
- "We sliced our Google Analytics data every which way, and these 4 patterns emerged."

All are examples of a common fallacy that I'm dubbing the "Pattern-Seeker."

You probably laugh at Nostradamus, yet it's likely you're committing the same error with your own data.

PATTERNS IN CHAOS

It's commonly said that basketball players are "streaky"—they get on a roll hitting 3-pointers (have a "hot hand") or develop a funk where they can't seem to land a shot ("gone cold"). These observations are made by fans, announcers, pundits, and the players themselves.

In 1985 Thomas Gilovich tested[1352] whether players really did exhibit streaky behavior. It's simple—just record hits and misses in strings like: HMHHMMMMHMMHH, then use standard statistical tests[1353] to measure whether those strings are typical of a random process, or whether there is something more systematic going on.

Turns out players are *not* streaky;* simply flipping a coin produces the same sort of runs of H's and M's. The scientists gleefully explained

* *Editor's Note:* Eight years after this article was published, a critical analysis of the Gilovich data suggested that perhaps there *is* such a thing as the "hot hand." Others joined the fray; the final result is inconclusive. More from Wikipedia.[1354]

"He can play like this, but he'll only be able to give 105%."

credit 1351

this result to basketball pundits; the pundits remained unimpressed and unconvinced. (Surprised?)

So they tried the same experiment backward: They created their own strings of H's and M's with varying degrees of true streakiness and showed those to pundits and fans, asking them to classify which were streaky and which were random. Again they failed completely, much like the "star field" test at the top of the article.

We perceive patterns in randomness, and it extends beyond casual situations like basketball punditry, plaguing us even when we're trying to be intentionally, objectively analytical.

Take the "interesting statistic" given by the baseball announcers in the first example above. Sure the last 5 of 7 batters were retired, but the act of **picking the number 7 implies that number 8 got on base, and so did the one before the sample window.** So it's at least "5 of

What we *can* conclude, is that it's extremely difficult to know whether something is random, even when you're a paper-writing statistician.

9," not "5 of 7," but that doesn't sound as impressive, even though it's the same data.

Unlike the basketball example, the baseball announcer's error runs deeper, and following that thread will bring us to our own marketing data and the heart of the fallacy.

SEEKING COMBINATIONS

Baseball statisticians record a dizzying smorgasbord of statistics that announcers eagerly regurgitate. Maybe it's because baseballers are a little OCD (evidenced by pre-bat and pre-pitch rituals) or maybe it's because they need something to say to soak up the time between pitches, but in any case the result is a mountain of data.

Announcers exploit that data for the most esoteric of observations:

> "You know, Rodriguez is 7 for 8 against left-handed pitchers in asymmetric ballparks when the tide is going out during El Niño."

This is the epitome of Pattern-Seeking—combing through a mountain of data until you find a pattern.

Some statistician examined millions of combinations of player data and external factors until he happened across a combination which included a "7 of the last 8," which sure sounds impressive and relevant. Then he proudly delivered the result as if it were insight.

So what's wrong with stumbling across curious observations? Isn't that how you make unexpected discoveries?

No, it's how to convince yourself you've made a discovery when in fact you're looking at pure randomness. Let's see why.

Perhaps the best example of this is the famous Bible Code[1355]—a "discovery" (turned into a best-selling book) that all sorts of predic-

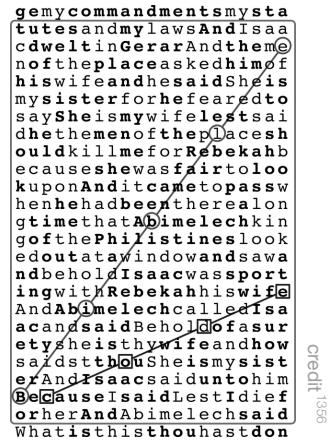


Figure 1

tions have been cleverly hidden in the Bible. We're supposed to then be excited with the tantalizing prospect of finding new predictions (though of course none were proffered).

The encoding worked as follows: Strip out everything that's not a letter (e.g. punctuation, whitespace, and so on), and arrange the letters in a grid. Then, look for works encoded along regular intervals of rows and columns in that grid. For example, Figure 1.

Words might be found at any interval, with any sized grid, so we'll have to experiment with many combinations. When we do, we find all sorts of words and phrases that match things that happened in the past two thousand years.

At this point, you should already see the problem. A huge set of letters, in a huge set of combinations, will automatically create words that you recognize. Indeed, why didn't we mention which translation we were using? Why does this work in English when it wasn't originally written in English? Because none of it matters.

Indeed, when you look for encoded messages in *War and Peace* using the same methods, you also get coded messages.[1357] All you need is a lot of letters and combinations.

Not unlike your A/B tests and combinations of "patterns" in your web analytics software.

EVEN A FAIR COIN APPEARS UNFAIR IF YOU'RE PATTERN-SEEKING

The fallacy is clearer when you look at an extreme yet accurate analogy.

I'm running an experiment to test whether a certain coin is biased. During one "trial" I'll flip the coin 10 times and count how often it comes up heads. 5 heads out of 10 would suggest a fair coin; so would 6 or even 7, due to typical random variation.

What if I get 10 heads in a row? Well a fair coin *could* exhibit that behavior, but it would be rare—a 1 in 1024 event. So if my experiment consists of just one trial and I get 10 heads, the coin is suspicious.

But suppose I did a "10-flip trial" one thousand times. A fair coin should still come up heads 3-7 times in each trial, but every once in a while it will come up 9 or 10 times. **Those events are rare, but I'm flipping so much that rare events will naturally occur.** In fact, in 1000 trials there's a 62% chance that I'll see 10 heads at least once.

This is the crux of the fallacy. When an experiment produces a result that is highly unlikely to have happened by chance, you conclude that something systematic is at work. But when you're "seeking interesting results" instead of performing an experiment, highly unlikely events will *necessarily* happen, yet still you conclude something systematic is at work.

Indeed, this is one reason why scaling a business is so hard (p. 773): Scale makes rare things common (p. 1345).

BRINGING IT HOME TO MARKETING AND SALES DATA

Let's apply the general lesson of the coin-flipping experiment to Google Analytics. There's a hundred ways to slice and dice data, so that's what you do. If you compare enough variables enough ways, you'll find some correlations:

> "Oh look, when we use landing page variation C along with AdWord text F, our conversion rate is really high on Monday mornings."

Except you sound just like the baseball announcer, tumbling combinations of factors until something "significant" falls out.

Except you're running 1000 coin-flip trials, looking only at the trial where it came up all heads and declaring the coin "biased."

Except you're seeing streaks, hoping that this extra-high conversion rate is evidence of a systematic, controllable force.

THE SOLUTION: FORM A THEORY, TEST THE THEORY

You can use apparent patterns to form a theory; that's good.

> *"The most exciting phrase in science isn't "eureka," but rather, "that's funny."*
>
> *–Isaac Asimov*

But then you have to test it, so that it's science, not baseball announcing.

So:

- Instead of using a thesaurus to generate 10 ad variants, decide what pain-points or language you think will grab potential customers and test that theory specifically (p. 913).
- Instead of rooting around Google Analytics hoping to find a combination of factors with a good conversion rate, decide beforehand which conversion rates are important for which cohorts, then measure and track those only.
- Instead of asking customers leading questions or collecting scattered thoughts, form hypotheses and intentionally test them (p. 239).
- Instead of blindly following the startup founder who dramatically succeeded (the 1-in-1000 coin flip? (p. 449)), gather advice and observations that align with your style and goals (p. 751).

And then never stop testing your theories, because you never know when the environment changes, or you change, or you find something even better.

> *"There are two possible outcomes: if the result confirms the hypothesis, then you've made a measurement. If the result is contrary to the hypothesis, then you've made a discovery."*
>
> *—Enrico Fermi*

Chapter 140:
Telling the 800-lb Gorilla to Shove it up his Ass

TREE-SHREWS · DON'T BEAT GOOGLE · NICHES
CAN'T FOLLOW · CAN'T DO

Every founder frets about competition from a big company, me included.

We scoff at their inability to innovate and for prioritizing shareholders over customers, but still we quiver in fear.

Dozens of people on Answers.OnStartups[1359] ask about it so I know I'm not alone. It always goes like this:

> I'm just a two-person operation with no budget. What if a huge company with a hundred software developers and a million dollars in marketing budget decides to copy my idea?

Answer: You're dead! Give up! No small company has ever survived competition with a large one!

Oh wait, that's not true. But poking fun doesn't help; maybe this article will.

credit[1358]

"What happens to the stock when I do *this*?"

First, take a deep breath and remember that **every little software company on Earth in under this threat**. This fact alone means competition—or threat of competition—isn't fatal, and possibly not even important.

DON'T FEAR THE DINOSAUR, FEAR THE QUIVERING WARM-BLOODED TREE-SHREW

65 million years ago an iridium-enfused extra-terrestrial **meatball o' death** caused what we would nowadays call a "disruptive market event," and the cold-blooded dinosaurs couldn't weather the shitstorm.[1360] It was the little cenozoic warm-blooded agile (oh sorry, now we're saying "lean") rodents who adapted by getting "outside the nest" to discover

how to eat cockroaches, because we all know that cockroaches are the one form of life that can survive anything.

Like unadaptable dinosaurs, whatever your large competitor is doing now is probably what they'll be doing two years from now, possibly four. Same message, same product, same pricing, and still taking a dump on Facebook instead of playing by the new rules. By then where will you be?

That kind of competition isn't scary. What *is* scary is another scrappy, smart startup like yours—another tree-shrew. The one who silently observes you from afar, then drives down the road you paved, skipping the mistakes you made and copying the good parts.

Take all your angst about big competitors and refocus it on the little ones. (I'll talk about this sort of competition in future post.)

YOU'RE SCRATCHING OUT A LIVING, NOT "BEATING GOOGLE."

If your only conception of "success" is to utterly destroy large companies, then I guess you should stop reading now.

But if you want to build a solid company, something you're proud of, something that pays handsomely (p. 45) but doesn't have to be worth $1B, then the game isn't "us or them." The question is: How can you own your little piece of the world; Not: How can you wrest $100m of revenue from a big guy.

It's not your purpose to "beat" another company. It's your purpose to define yourself on your own terms, not in terms of how you're like or unlike someone else.

Sure it's constructive to "set your sights" on a competitor, actively trying to beat them in the marketplace or even steal their customers (e.g. give a discount if someone switches to you). But ultimately the

only thing that matters is that you earn more and more customers, whether or not anyone else does too.

USING A GORILLA TO INCREASE YOUR OWN PROSPECTS

It can actually be an advantage to have a big player in your market, especially if they enter your market after you're established.

At Smart Bear[1361] we make a peer review tool for software programmers; you don't have to be a geek to know that any software development tool company shares the following fear: "What if Microsoft copies us?" But we know that any code review tool from Microsoft would work only with their own version control system and only inside Visual Studio. (Can you imagine a tool from Microsoft that supported ClearCase, ran inside Eclipse, and had excellent support for Java?)

So what if they did copy us, and what if as a result they owned 100% of the Visual Studio market? **Well that still leaves every other market on Earth**. And then all of Microsoft's competitors would also need a code review tool so they don't fall behind on the hallowed competitive analysis chart, so suddenly IBM, CA, Oracle, Serena, CompuWare, and HP would need a code review tool *right away*. What better way to accomplish that than to buy the #1 (or maybe now #2) code review tool company—hey that's us!—which by the way is profitable at a time when any company is happy to have a department that's generating cash.

In short, Microsoft copying the idea would only validate the market, causing the value of Smart Bear to increase.

What actually happened is instructive too: Microsoft added the concept of "shelving" and put the absolute least amount of effort into supporting code review (it's literally a check-box that indicates that,

somehow, somewhere, a code review happened), so the result is that we sell a ton of Code Collaborator to Visual Studio shops.

In other words, they validated the market by entering it, but exactly because they're a huge company they couldn't make it good enough to stop us.

GO WHERE THEY CAN'T FOLLOW

Big companies play only in big markets.

It's logical: With all the expensive machinery and bureaucracy it takes a dump truck of money and dozens of man-years to build something new, so the opportunity has to be enormous. Even if they *were* successful in a small market there wouldn't be enough profits to move the big needle at the top.

So Microsoft can't attack a market unless there's a potential to earn at least $1B. But wouldn't *you* be happy playing in a market where you'd be able to rake in "only" $100M? Of course you would.

I'm not talking about carving out micro-niches where only seven people on Earth are potential customers. Just don't go after massive, general markets like "everyone with a digital camera" or "anyone with a smart phone" or "all software developers."

DO WHAT THEY CANNOT

Big companies have significant advantages like money, a brand, a team, and a large customer base, many of whom will never switch even when presented with a clearly-better alternative.

credit 1362

"I'm looking for something in a small pond."

Their brand alone is a powerful force you probably cannot overcome, e.g. "eBay is trustable" or "Apple is cool" or "IBM is safe."

But the same attributes which deliver those advantages are also restrictive:

- They can't release a completely-revamped, brand-new version because they can't retrain 200,000 users.
- They can't take a risk because protecting the existing revenue stream is more important than *anything* else, even if it means their ultimate demise.
- They can't quickly convert new ideas to released code because there's requirements and documents and designers and approvals and schedules and testing and vetting.
- They can't change their image because there's too much momentum with the old one. For example if they have a reputation for bad tech support, even if it gets remedied most people will still think of them has having bad support.
- They can't observe and react quickly to changing market demands because there's too many layers of people and process, and too

> many people whose careers depend on maintaining the status quo.

For example, Intuit needs to look solid and timeless, their developers know C++ and desktop applications, and they can't retrain the computer-phobic home users of Quicken... so they cannot create Mint.[1363]

As another example, IBM requires expensive infrastructure, development teams, and sales channels to command multi-million dollar consulting deals, but that also means it's not profitable to do a small deal, which means small consulting shops never worry about competing with IBM.

They can't change their product, so you can innovate without competition. They can't change their image, so you can fill the gap. They can't listen to a customer and make an impact one week later; you can.

Do what they can't do (p. 295), be what they're not, and you won't have to worry about competing on those points.

Chapter 141:
Tech Support is sales

THE FACE OF YOUR COMPANY
TECH SUPPORT IS SALES · A PLEASANT SURPRISE
OUTSIDE, INSIDE

You probably think of "tech support" as the bottom of the food chain. "Shit flows downhill" and all that. After all:

- Tech support deals with insane customers.
- Tech support answers the phone; a job even sales people don't want.
- Tech support keeps angry customers at bay while having no power to effect change.

Yep, that sounds lowly. Dismal too—how would you like to deal with an irate voice screaming at you when you *know* how to fix the problem but lack the authority to do it? This is a masochistic job for a poor slob with no other job prospects, right?

Impressive, until he opens his mouth. …maybe like your company

If this is your attitude, your conception of tech support is completely backwards and you're missing out on important channels for marketing, product development, and sales.

THE FACE OF YOUR COMPANY

We've all been jarred by someone's voice not matching their picture. Take English footballer David Beckham, the quintessential picture of manly sportif—washboard abs, ex-captain of the English national team, and married to Posh Spice.

But then he opens his mouth. It's like Kermit the Frog got kicked in the balls. (Oh, sorry UK folks, I meant kicked in the *bollocks*.) It's so unexpected it's the only thing you remember. Of the 3,204,523 pub conversations where someone said "Have you heard him *speak*?" maybe only 17 could tell you what he actually *said*.

You assume your home page is the public face of your company, but what happens when you open your mouth? What happens when your bullet points collide with your behavior?[1364]

For most of your customers, **tech support is the only human interaction they'll have with you.** Are you really going to leave that up to your worst-treated, least-paid, least-qualified employees?

TECH SUPPORT *IS* SALES

At Smart Bear we made millions of dollars in both individual and enterprise sales without "sales."

Well, at least without the usual definition of "sales"—a collection of processes, personalities, and management single-mindedly focused on hauling in revenue on a quarterly schedule.

How did we get six-figure deals without playing golf or using Salesforce? Simple: Our tech support *was* sales.

You could say the purpose of tech support is to answer questions or to unstick people who are confused, but I say **the purpose of tech support is to make your customers fantastic at their jobs**, which happen to involve your product. (Yes, I'm flagrantly paraphrasing the legendary Kathy Sierra, but the idea applies as much to tech support as to product development.)

So this means you don't just help them locate a command in the menubar, you find out what they're trying to accomplish and help them do *that*. You don't just explain a feature but help them use the result to impress their boss. You don't just apologize because you don't have the feature they want, you help them work around it and be successful anyway. You know your product and problem-space better than your customers, so it's not that hard to make them far more successful than they would be stumbling around without calling tech support.

Enabling your customers isn't just about your *product*, but rather your *entire company*. Make your customer awesome and she'll give you money so she can keep being awesome.

That's sales.

A PLEASANT SURPRISE

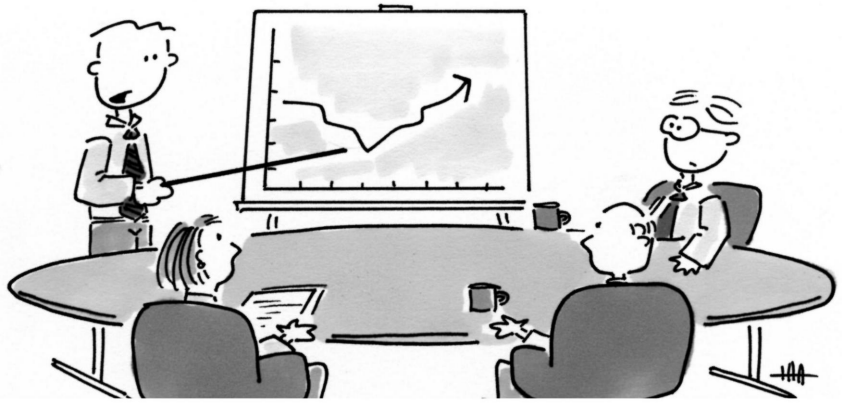
Everyone's stereotype of tech support is negative. Oh the tales:

- Ask tech support how to change the font and they'll tell you to reboot your laptop.
- Ask tech support to change your billing address and they up-sell you on three things you don't want.
- Calling tech support requires a GPS to navigate the labyrinth of menu options (which may have changed), wait-queues, and typing in your account number 3 times "for security purposes," as if someone who stole your account number is incapable of typing it more than once.

When your customers expect a turd sandwich and you deliver a turkey club with chipotle aioli, you earn major bonus points, like users twittering about your service, people switching to your service [1366] because of tech support, or customers not only following your Tweets but instructing *their* followers to do the same.[1367]

Oh look! Apparently tech support is a better "social media outreach" program than hiring interns to spray comments on random blogs. Are you surprised?

They say "under-promise, over-deliver," and tech support has "under-promise" built in! Sure super-fantastic tech support is best, but



credit[1365]

"So, as you can see, customer satisfaction is up considerably since phasing out the complaint forms."

even if you merely act like a human being you're already ahead. If you just answer email with a non-automated response you're killing it.

Why pass up such an easy opportunity to thrill a customer? Isn't "a pleasant surprise" too rare in business, and don't you want to be known as the company where it happens every day?

THE CLOSEST THING TO GETTING "OUTSIDE THE BUILDING" WHILE STAYING INSIDE THE BUILDING

The Internet is abuzz with Steve Blank's phrase that everything you need to know about your customers is "outside the building,"[1368] meaning that real customer development means talking to folks face to face, seeing their problems in the wild, and watching their faces react to your pitch, not brainstorming around a whiteboard and twiddling the font size in your PowerPoints. And I agree!

Still, for the Work-a-preneur[1369] or the bootstrapper with no travel budget it's hard to get outside the building. Yes you should try as much as you can—it's worth it—but what about the other 94% of the time that you're at your desk, by which I mean the coffee shop table closest to the power outlet that isn't loose?

Tech support is the next-best thing. Tech support is where people complain about what's not working, what's missing, and what's confusing. But it's not enough to just catalog problems!

The insights lurk in the meta-questions. If someone's confused, for example, the immediate task is to set them straight, but there's valuable product development to be had:

- What caused the confusion in the first place?
- Is my customer's world-view (p. 239) different from mine?

- Is our terminology wrong?
- Are we using the wrong metaphors?
- Do I need to optimize the new-user experience instead of the expert-user experience?

Those are tactical questions stemming from the immediate problem, but then there's even more interesting strategic questions:

- Does this hiccup belie a customer pain-point I didn't know existed but I can solve?
- Is there enough evidence of a conceptual mismatch that I should pivot?
- Is there a new product idea (p. 793) here?
- If they're abusing my product to get what they really want (p. 193), can I provide what they really want from the start?

This last line of questioning is exactly how Smart Bear came to be (p. 9) a company about peer code review and not "version control data mining." If I hadn't paid attention to these meta-questions, you wouldn't be reading this right now. Yes, it's that critical.

To answer these you have to go back and forth with customers to hack into the root cause. You have to see hundreds of emails so you get a gut-feel for what customers are experiencing—something you can't get from a Incident Summary Report or some-such automation.

Tech support is the closest, most honest chance for product development—certainly more straightforward than squeezing it through traditional "sales." Here's where real users discover and report on your product.

Are you listening, or just throwing it away?

Chapter 142:
Not disruptive, and proud of it

EXPLANATION · DISTAIN · ACCIDENTAL
UNPROFITABLE · LIKELIHOOD

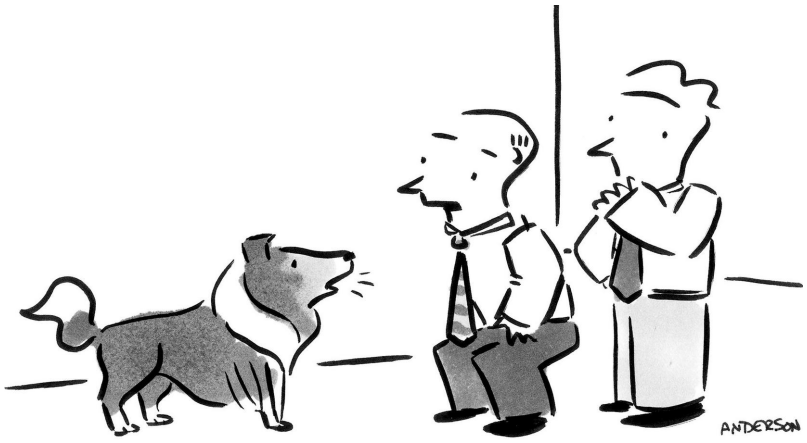
The phrase "paradigm shift" died during the 2000s tech-bubble along with "portal" and "think outside the box," and yet the underlying concept never died. For the past ten years it's been called "being disruptive."

When I get pitched—usually by someone raising money—that they "have something disruptive," a little part of me dies. You should be worrying about making something useful (p. 71) and delightful (p. 275), not about how "disruptive" you can be.

Disruption is occasionally a consequence of innovation, but when it becomes your primary purpose, you've lost sight of why you should be doing any of this in the first place.

"Disruptive" is the in-vogue word for the opposite of "incremental improvement." A disruptive product causes such a large market shift that entire companies collapse (the "dinosaurs" who don't "get it") and new markets appear.

Disruptive is fascinating, disruptive changes the world, disruptive makes us think. Disruptive also sometimes (p. 419) generates billions



credit[1370]

"What's that boy?! A paradigm shift?!"

of dollars, which is why venture capitalists have always loved it and always will (and they're not wrong, given their goals).

But disruptive is rare and usually expensive. It's hard to think of disruptive technologies or products that didn't require hundreds millions or even billions of dollars of investment, spending ahead of revenue. Most of us don't have access to those resources, and many of us don't care, because we'd rather work on an idea we actually understand and can build ourselves, an idea that might make us a living, that people want and need (p. 259), maybe even something people love (p. 275).

There's nothing wrong with incremental improvement. What's wrong with doing something interesting, useful, new, but not transcendental? What's wrong with taking a known problem in a known market (p. 71) and just doing it better or with a fresh, unique perspective (p. 543)? Do you have you create a new market and turn everyone's assumptions upside down to be successful? **Should you?**

I'm not so sure. This is the argument against disruption.

IT'S HARD TO EXPLAIN THE BENEFITS OF DISRUPTION

Have you tried to explain Twitter someone?* Not the "140 characters" part—the part about why it's a fundamental shift in how you meet and interact with people?

Hasn't the listener always responded by saying, "I don't need to know what everyone had for lunch. Who cares? What's next, 'I'm taking a dump?'" They don't get it, right? But it's hard to explain.

There *are* ways to elucidate the utility of Twitter, but even the good ones are lengthy and require that listeners have patience and open minds—two attributes in short supply.

"It's hard to explain" means it's a terrible sales pitch. "You just need to try it" and "trust me" don't cut it. That may be OK for Twitter —today—but what about the 100 other social-networking-slash-link-sharing networks that didn't survive? Ask them about selling intangible benefits.

PEOPLE DON'T WANT TO BE DISRUPTED.

If you're reading this you're probably more open to new ideas and new products than most, because you're inventing a new product, starting a company, or you're just ruffled because I'm pissing on "disruptive"

* *Editor's Note:* This was written in 2010, when Twitter was still relatively new. In retrospect, the point was correct—the vast majority of people who eventually got obsessed with social media, didn't get obsessed with Twitter.

and you're looking for nit-picky things to argue with me about. (Hi Hackernews comment section!)

But most people are creatures of habit. **They don't want their lives turned upside down.** They launch into a tirade of obscenities if you just rearrange their toolbar. When they hear about a new social media craze they cringe in agony, desperately hoping it's a passing fad and not another new goddamn thing they'll be aimlessly paddling around in for the next decade (Figure 1).

Change is hard (p. 1299), so a person has to be experiencing real pain to want to change. Selling a point-solution for a point-problem is easier than getting people to change how they live their lives. Identify-

Figure 1

ing specific pain points and explaining how your software addresses those is easier than trying to tap into a general malaise and promising a better world.

TECHNOLOGY WE NOW CONSIDER "DISRUPTIVE" OFTEN WASN'T CONCEIVED THAT WAY.

Google was something like the 11th major search engine, not the first.[1372] Their technology proved superior, but "a better search engine" was hardly a new idea. In retrospect we say that Google transformed how people find information, and further, how advertising works on the Internet.

Disruptive in hindsight, sure, **but the genesis was just "better"** than the 10 search engines that came before. (Or 18.[1373])

Scott Berkun[1374] gives several other examples in a recent BusinessWeek article.* He highlights the iPod—an awesome device, but not the first of its kind. Rather, there were a bunch of crappy devices that sold well enough to prove there was a market, but no clear winners. Here an innovation *in design alone* was enough to win the market. Not inventing new markets, not innovative features, not even improving on existing features like sound quality or battery life—just a better design, unconcerned about "disrupting" everything else.

Setting your sights on being disruptive isn't how quality, sustainable companies are built. Disruption, like expertise, is a side-effect of great success (p. 1529), not a goal unto itself.

* It has since been taken down; the perils of writing for decades on the Internet.

THE DISRUPTORS OFTEN DON'T MAKE THE MONEY.

The construction of high-speed Internet fiber backbones and extravagant data centers fundamentally changed how business is conducted world-wide both between businesses and consumers, but many of the companies who built that system went bankrupt during the 2000 tech bubble, and those who managed to survive have still not recovered the cost of that infrastructure. They were the disruptors, but they didn't profit from the disruption.

Disruptive technology often comes from research groups commissioned to produce innovative ideas but unable to capitalize on them. Xerox PARC invented the fax machine, the mouse, Ethernet, laser printers, and the concept of a "windowing" user interface, but made no money on the inventions. AT&T Bell Labs invented Unix, the C programming language, wireless Ethernet, and the laser, but made no money on the inventions.

Is it because disruptors are "before their time," able to create but not able to hold out long enough for others to appreciate the innovation? Is it because innovation and business sense are decoupled? Is it because "version 1" of anything is inferior to "version 3," and by the time the innovator makes it to version 2 there are new competitors—competitors who don't bear the expense of having invented version 1, who have silently observed the failures of version 1, and can now jump right to version 3?

"Why" is an interesting question, but the bottom line is clear: Disruption is often unprofitable.

SIMPLE, MODEST GOALS ARE MOST LIKELY TO SUCCEED AND MAKE US HAPPY.

It's not "aiming low" to attempt modest success.

It's not failure if you "just" make a nice living for yourself. Changing the world is noble, but you're more likely to change it if you don't try to change everything at once.

I made millions of dollars (p. 45) at Smart Bear with a product that took an existing practice (peer code review) and solved five specific pain points (annoyances and time-wasters). Sure it wasn't worth a hundred million dollars,* and it didn't turn anyone's world inside-out, but it enjoys a nice place in the world and it is incredibly fulfilling to see people happier to do their jobs with our product than without it.

Had I tried to fundamentally change how everyone writes software, I'm sure I would have failed.

I made less money personally at ITWatchDogs,[1376] but the company was profitable and sold for millions of dollars. We took a simple problem (when server rooms get hot, the gear fails) and provided a simple solution (thermometer with a web page that emails/pages you if it's too hot). There were many competitors, both huge (APC with $1.5 billion market cap), mid-sized (NetBotz with millions in revenue and funding), and small ($2M/year operations like us). We had something unique—an inexpensive product that still had 80% of the features of the competitors—but nothing disruptive.

Had we tried to fundamentally change how IT departments monitor server rooms, I'm sure we would have failed.

---

* Ten years after this was written, Smart Bear continued to grow and acquired other profitable, bootstrapped companies in the software-quality space, eventually being sold for nearly $2B.[1375] None of the products were "disruptive," but all were well-executed.

There's nothing wrong with modesty. Modest in what you consider "success," and modest in what you're trying to achieve every day:

> *My daughter convinced me that insisting something be Deeply Meaningful With Purpose can sometimes suck the joy from it."*
>
> *–Kathy Sierra*[1377]

Of course it's wonderful that disruptive products exist, improving life in quantum leaps. And it's not wrong to pursue such things!

But neither is it wrong to have more modest goals, and modest goals are much more likely to be achieved.

Editor's Note from 2024: In the fifteen years since this was written, I started another company WP Engine, which is still growing and profitable at hundreds of millions of dollars in revenue, powering 2.5% of the largest websites in the world. It still wasn't because we wanted to "fundamentally change how website are created," but rather we saw how to dramatically improve something that people were already doing by the millions: Running WordPress sites. More of its origin story is here *(p. 9)**.*

Chapter 143: Maybe not so much with the "optimization"

In the never-ending quest for optimization, A/B tests, metrics, and funnels, we're in danger of losing the fun and value of creative work.

When we demand overwhelming customer outcry before committing to the slightest product change, we're in danger of losing the value of creating a cool feature that takes too much effort but people just love (p. 101).

When we do the minimum necessary to get the job done, we're efficient but not thrilling. We're "lean" but we're not stirring hearts. We're effective but not playful.

I'm as excited as everyone else about Lean principles gaining traction, and sure most companies are erring on the side of too little objective feedback (p. 1541) rather than too much. Still, every

This weird-looking thing is a result of discontinuous innovation, not incremental improvement.

article I read turns the creative process of business and product design into Vulcanian objectivity.

Sometimes, you should do something just because it's cool. There's such a thing as product "taste."

Look at this incredible display of affection IHumanable has for his computer:[1378]

> This is one of the reasons I love my new iMac, **it's just a beautiful magic floating screen filled with win.**

You couldn't ask for a stronger endorsement. This is even better than "It saved me $725,231." This is beyond utility—**this is love**. (Love wins.) (p. 275)

Does love come from feature bullet points? Do you earn love through A/B tests and implementing features off the top of GetSatisfaction? Or is this something else, something deeper, **something less incremental, less data-driven, more gut feel, more emotional?**

My first product at Smart Bear[1379] had a non-optimal, floating-in-win invention called the "mini-viewer." Here's its story.

Code Historian was my first product. It was the first file difference viewer with built-in support for version control systems, letting you view various historical versions of a file side-by-side. You could switch between which versions you were comparing with one click (Figure 1).

The thing to focus on is that user interface element in the bottom-right corner. That's the "mini-viewer," and **in every measurable sense it's a terrible business decision.**

The mini-viewer summarized the modifications—the lines added, changed, and removed—so the user could easily see how many changes there were and where they're located. Sounds useful, right?

Right, except it's a really wasteful, expensive way to do it. Many competitors used a different technique I call "boogers," because to me it looks like someone shot snot rockets all over the screen, and also because it's fun to deride competitors, because it feels good to make fun of other people who (appear to) have more revenue than you do.

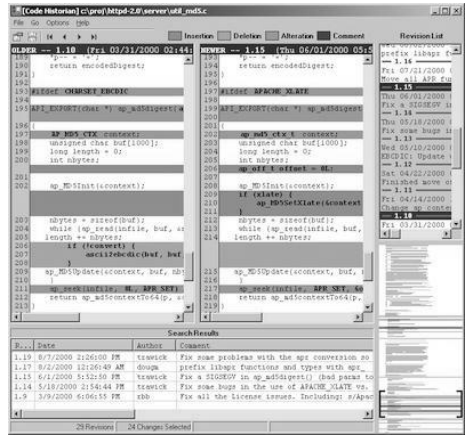
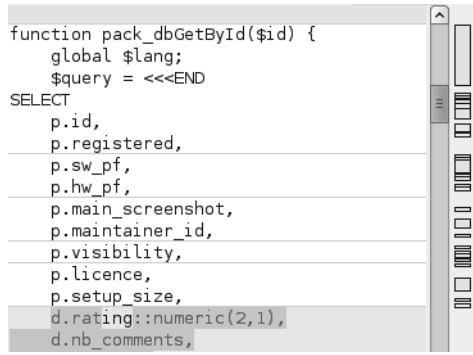


Figure 1



```
function pack_dbGetById($id) {
    global $lang;
    $query = <<<END
SELECT
    p.id,
    p.registered,
    p.sw_pf,
    p.hw_pf,
    p.main_screenshot,
    p.maintainer_id,
    p.visibility,
    p.licence,
    p.setup_size,
    d.rating::numeric(2,1),
    d.nb_comments,
```

Figure 2

But don’t you agree? They look like boogers (Figure 2).

The boogers are smeared by the scrollbar, indicating where you’d need to scroll to see differences between the two versions of the file.

Now by all of the usual arguments for Lean, Agile, and minimalism, I should have used boogers too:

1. Boogers were already semi-standardized. User interfaces should follow the principle of "least surprise"—if people are used to a certain metaphor, icon, or behavior, you should honor that so people understand your product immediately. No one else had a mini-viewer.
2. Boogers occupy minimal screen real-estate. It's just a thin strip no wider than a scrollbar; in fact some products put the boogers *on top of* the scrollbar. The mini-viewer is not only larger, it has significant width, which means you have to occupy the rest of the right side of the screen with other crap.
3. Boogers appear right next to the scrollbar, which is where you look anyway when navigating the file.
4. Boogers take less effort to compute than the algorithm for determining color variations in the mini-viewer.
5. Boogers take less effort to draw. Boogers are drawn on the screen once, and don't change unless the window is resized—an infrequent operation. The mini-viewer however indicates your current scroll position in the file (those black brackets) so when you're scrolling around the file the speed at which you can recompute and redraw the mini-viewer matters. Drawing directly on the screen causes flickering, so you need off-screen buffering. In short, **the mini-viewer is a lot more programming effort with a lot more chance for bugs.**
6. The mini-viewer doesn't convey more information than boogers do.

And yet, everyone loved the mini-viewer. People sent emails saying they used Code Historian just because of the mini-viewer. Some developers wrote in asking how I was able to render it so efficiently. It was always a high point in product reviews.

The mini-viewer was wasteful, but fun. It wasn't optimal and had no measurable benefit to usability, but it was "filled with win." **It took extra effort but it was endearing**—an important attribute not easily captured with metrics and spreadsheets.

Now sure, there are many of aspects of business and product development where it's best to stop obsessing and just cut corners. Often we can and should accept 80% of the benefit if it means 20% of the effort. Customers generally prefer the *right* features over *more* features.

But sometimes it's your job to fill the screen with **joyous win**.

Chapter 144:
A butterfly flaps its wings and you make a sale

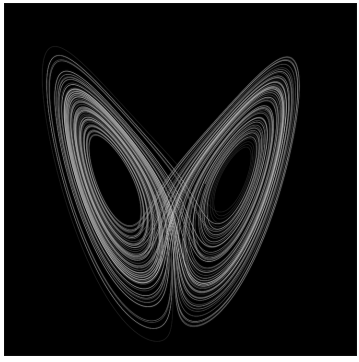
"So you believe that when you flap your wings on one side of the planet, all kinds of crazy things are caused on the other side? That sounds like a lot of responsibility."

credit 1380

It's easy to be taken in by the idea of the Butterfly Effect:[1381] That a butterfly gently flapping its wings in the jungles of Madagascar eventually causes a Typhoon off the coast of Jakarta.

Or, updating for modern-day relevancy, Naomi Dunford[1382] pounds a curse word into a WordPress and Brian Clark[1383] makes $172. Or Dave McClure[1384] releases a silent-but-deadly[1385] outside a Menlo Park Starbucks and a social media company gets funded in Boston.

It's a great story: Little actions can have enormous influence. A small favor you do on Twitter results in a viral post seven months later. A small change to your download page results in 20% more trials. A subtle shift in background color increases average time-on-site by 27 seconds.



credit[1386]

Lorenz Attractor

We're *willing* to believe it because mathematicians have shown that it *is* true for certain types of complex-yet-sensitive systems like weather and economics and fractals.

We *want* to be believe it because it's harmonious and comforting to think that everything is connected—that the tiniest action has the potential for significant effect. It means every person has the power to change the world.

> *"Even the smallest person can change the course of the future."*
>
> *—The Fellowship of the Ring, J.R.R. Tolkien*

For me the most compelling evidence comes from cognitive psychology where studies abound with astounding tales of subtle environmental changes radically and systematically affecting people's behavior.*

It's relevant for marketing and sales because it's an inside scoop about how to manipulate strangers on the sly. It's akin to subliminal messaging, but more pervasive, more powerful, and less susceptible to biting satire.

Eerie examples:

- Touching merchandise while you're shopping increases the chance that you'll purchase it. (source[1388])
- Students performed word-searches from random words. Some of the puzzles were seeded with words associated with old age, e.g. gray, wrinkle, bingo, Florida. While traversing a hallway after solving the puzzle, those students given "old" words walked more slowly. (source[1389])
- Students took a survey about health risks; half walked down a hallway where someone was sneezing. Those who passed the "ill" confederate reported a more negative view of the American health system and believed the average American was more likely to die of heart attack. (source[1390])
- Students were asked to recount memories while moving marbles between two trays. When moving marbles from a lower tray to a higher one, the memories were more positive; when moving downward the memories were more negative. (source[1391])

This news should be simultaneously titillating for marketers ("Ooo, puppet strings!") and frightening for consumers (Are you *ever* in control of your own decisions?).

---

* *Editor's Note:* 14 years after this was written, in 2024, most of these "results" have been debunked[1387] because they're impossible to reproduce. This alone should cause us to be skeptical of their application in marketing, but read on for another reason, even for the (unknown) subset of them that are true.

The Butterfly Effect
(Actually Rorschach, 1948)

But actually, when taken to its logical conclusion, **you have to ignore it**.

After all, these studies touch a tiny subset of the hailstorm of influences constantly befalling us. When I'm at the mall I'm passing people who are coughing just like the experimenter in the study... but also people who are angry, laughing, sitting, jogging, yelling, sleeping, shoplifting, and eating. Each storefront beckons me with colors, shapes, fonts, compositions, arrows, borders, lighting, and even sounds and smells.

All this is (apparently) tugging me in different directions, just below the veneer of consciousness where my impulsive, subconscious lizard brain[1392] is eagerly lapping up the stimuli and directing my attention and my wallet.

But then again, despite these impressive efforts, I'm distracted by the P.A. system blaring about a 6-year-old knee-deep in the fountain outside the Men's Dillards. And then my cell phone goes off with a new tweet mentioning @asmartbear[1393] and my heart goes all aflutter *(Ooo, attention! Please love me so I can love myself!)*. And then a butterfly flaps its wings (this time in Argentina) and suddenly and inexplicably I decide against the indulgence of a Double Doozie Cookie.

If I'm constantly influenced in all directions, the forces cancel each other out, and none of them matter.

It's worse on the Internet. You're competing not only with the real world but with the virtual world of tabbed browsing, Twitter alerts, back buttons, bouncing tray icons, and instant messaging.

It seems to me that instead of chasing subtle subliminal effects, most of which will be wiped away by the ambient noise of life, **we could spend our time on the big-ass, in-yo-face, non-subliminal effects.**

Like, if you get a popular blogger to mention you, that's more influential than the color of your logo. *(200 words from Seth Godin is good for 10s of 1000s of unique visitors.)*

Like, if you have a compelling story that people intrinsically want to spread, that's more influential than building a snappy animation for your home page.

Like, if you thoroughly thrill one person in a product demo, that's more money in the bank than 1,000 people hitting your website and getting "branded" that you're "trustworthy" because of your steel-blue color palette and stoic font. *(I'll take one Tom (p. 1559) over ten thousand StumbleUpon hits.*[1394] *)*

I like the idea of subtle yet powerful influences as much as anyone else, and I'm not saying design and attention to detail isn't important or valuable (although maybe it isn't (p. 853)). I just think most of our efforts are drowned out by the seething distraction that is the Internet and life in general.

Take care of the big stuff.

Chapter 145:
Enough with the “expert” guilt

“You see any next generation leadership,
you call me.”

credit 1395

I'm sick of being admonished that success is predicated on spending the next 10,000 hours of my life becoming "an expert."

I'm sick of hearing about how I should be molding my life in the image of Michael Phelps or Albert Einstein, because the only thing that separates me from genius is identifying my strengths and working really really hard.

I'm calling bullshit.

We're so busy trying to make ourselves into outliers[1396] **that we're forgetting about what's important.**

Penelope Trunk pushed me over the edge when she wrote[1397] that for the last two years she's been schlepping around a Harvard Business Review article called "The Making of an Expert" because:

> "The article changed how I think about what I am doing here. In my life. I think I'm trying to be an expert.

Penelope goes on to equate being an expert to "success," and laments that she isn't an expert in anything, nor is she making headway.

I don't know whether this is funny or sad, because she wrote this on her blog—a blog with 48,767 subscribers (at the moment). There are literally a million people trying to be "expert" enough at *anything* to achieve that level of "success," and almost none of them will ever be that "successful."

Oh yeah, and this comes on top of a six-figure book deal and years of writing for teeny inconsequential publications like the Wall Street Journal and Time magazine.

But Penelope considers herself neither a success nor an expert.

Yeah, right. She *is* a success. In fact, don't you agree her problem isn't a lack of expertise but rather that she shares my irrational yet commonplace feelings of inadequacy (p. 457)?

If by her definition she's not even close to being an "expert," **clearly being an expert isn't required for being successful.**

She goes on to explain how much effort it takes before you're allowed "expert" status (my emphasis):

> "You need to spend at least **ten years** working in a **very focused, everyday** way on the thing you want to be great at. Evidence: high school swimmers today would beat Olympic records from years ago."

That's not "evidence." There are more high school swimmers than ever, therefore more opportunities to find and train great swimmers. They have access to diet, training, technology, and facilities that didn't exist years ago. That's what it is "evidence" of.

And anyway, supposing it does take that much sheer effort, clearly it also takes talent (though she denies this, as do other,[1398] *cough cough*, experts). I'm a case in point: I practiced the piano for an hour a day for more than ten years. I became good, but there were others who practiced twice as much who were worse, and still others who practiced less and are much better.

We all know this. Why are we allowing people to tell us otherwise?

Not one of the successful entrepreneurs I know *started* as an expert. Rather, **career and expertise were developed simultaneously**, eventually resulting in success when coupled with a few key events (due as much to luck as effort (p. 1035)).

Pick anyone. Sergey and Larry weren't advertising experts before they started Google. Joel Spolsky wasn't a blogging expert before starting FogCreek. I didn't know anything about peer code review before starting Smart Bear.

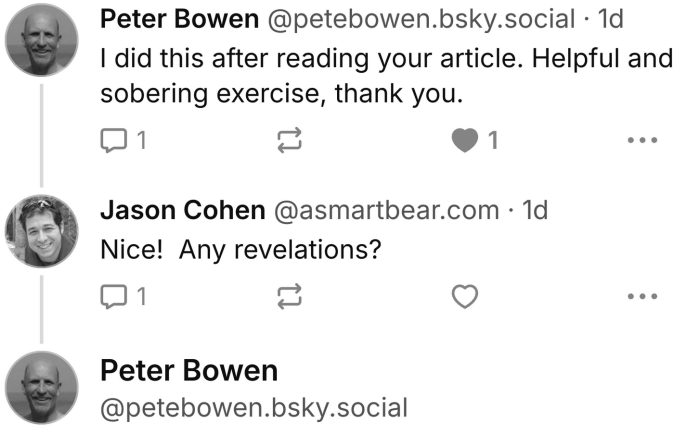
In fact, in all these cases it would have been impossible to have been an expert! Why? Because Google reinvented advertising, there were no "blogs" when Joel started posting essays, and there was no tool for code review until I invented one.

Innovation defies prior expertise.

So let's stop being distracted with these arbitrary definitions, artificial goals, and unnecessary prerequisites to "success."

Let's just get back to work.

Chapter 146:
Rude Q&A: The constructive devil's advocate



Found some real weak points in my offering that would be a red flag for me if I was a buyer. Also identified a type of person who would never buy so it makes it easier to develop the marketing material.

Nothing clarifies things quite like a hyperactive, all-knowing, all-seeing, real asshole of a devil's advocate beating the living crap out of you.

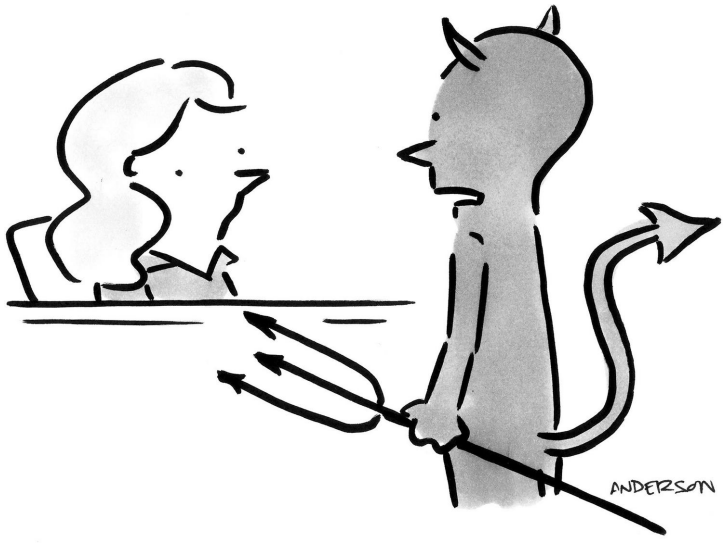
Baseball players swing heavy bats before going up to the plate; acclimating to difficult working conditions makes it easier to hit the ball out of the park.

What's the equivalent of heavy-bat-swinging for honing your skills at pitching your product and raising money for your company?

For years I've been a fan of Scott Berkun's concept of Rude Q&A:[1400]

> **What would the meanest, nastiest, but smartest people in the world grill you on when you show your work?**
>
> A "Rude Q&A" is a list of questions [about your work that] you don't want to hear.

When you're contemplating an exciting new idea, you don't want to hear questions that might contradict your concept.



credit[1401]

"I'm here about the details."

And of course, that's exactly when you need the biggest, baddest, smartest, devil's advocate to challenge all your assumptions.

It's not just about testing the mettle of your ideas, it also forces you to refine and clarify your marketing messages (p. 627), your target customer profile (p. 317), and your feature set. When you're being grilled there's no room for being generic (p. 1439) about how you're different (p. 1475) from the competition and where you are strongest (p. 543), no leniency for not knowing exactly what customer pain you solve (p. 239) and why they should love you (p. 275), and no clemency for wavering on your company values (p. 827) and what compromises you're (not) willing to make.

Scott goes on to explain just how unfair the questions need to be:

> **Make sure to include questions that are unfair or based on erroneous information.** Reporters, clients, and the public all have their share of unfair questions and erroneous information, and you want to be ready for them.
>
> These answers take more time as the responses need to be more polite and mature than the questions. They also need to carefully refute assumptions in the questions without being dismissive.

I love it; now we're deep into "heavy bat" territory.

So how do you go about writing your Rude Q&A? The hardest part can be coming up with the questions.

To get you started, I've assembled a laundry list of questions common to many startups:

- Your biggest competitor just dropped their price to $0. How do you continue to justify your price point?
- If your idea is any good, you'll soon have competition and even copycats from multiple players, both incumbents (p. 295) and new

startups, both funded and bootstrapped, both smart and stupid, both large and small. How will you persevere?

- If the economy stays bad for two more years, how will you survive?
- The last thing anyone needs is *another damn tool*. What's the overwhelming reason I should even bother looking at you?
- Competitor _____ is doing better than you (p. 1407). What are you going to do about it?
- Technorati reports one million new blog posts appear every day. Why should I read yours?
- If you raised prices 50% (p. 1125), would any customers stay? How are you going to 10x the quantity of those kinds of customers?
- What are the top three features your competitor has that you lack? How do you address that today, and what are you doing about it in the next six months?
- How can you call yourself an expert (p. 1529) when you've only been at this for a year?
- There's only two of you. What happens if someone leaves? The company is finished?
- What's the single most important thing you need to accomplish in the next 12 months, and how are you going to get it done? (Do you even know what it is?)
- Your company is going to top out of growth (p. 1153) soon because of your high cancellation. How are you going to fix the fact that so many people decide they don't like you, even after buying from you?
- What are three tangible, undeniable ways in which your product/company saves more money than you cost and saves more time than you consume?
- Truly great products and companies are rare, even when smart people are at the helm. What makes you think you have what it takes (p. 457)?

credit[1402]

"Also, would you mind taking your shoes off?"

- There are thousands of consultants who make the same claims you make: high-quality, on-time, on-budget, good service, happy customers. What makes you any different?

These are generic; you'll need to come up with more specific attacks. For example, if I were defending this blog and answering the question about why anyone should read it, I would make the question more specific:

> There are already too many blogs about startups, especially high-tech startups. Those blogs are far more popular than yours, their authors far more famous, and their advice is excellent. Smart Bear is a success but it's nothing like the success earned by someone like Steve Blank.[1403] Why should anyone listen to you?

And here's my answer:*

> I read those blogs; they're great!

There's two reasons why this one should exist:

1. For every Jason Fried who says[1404] "simple design is better than complex features," someone else needs to point out that they've (I've!) made millions with poor graphic design (p. 853) and too many features. For every Seth Godin who says a tribe of 1,000 followers is all you need,[1405] someone else needs to point out that it's not actually true in practice.[1406] When every piece of advice has a counter-argument, you realize that your goal is to find advice that matches your context, your goals, your personality, your constraints, your strengths, and that means we need blogs like this one, sourced from my own combination of those things, for the benefit of those for whom my specific attitude and experiences make them a better version of who they already are (p. 751).

2. Think about the best teacher or mentor you ever had, someone whose words have rung in your head forever after, or someone who changed your life, even if by accident. Did that person invent the subject they taught? Even 1% of it? No. Having a genuine life-changing impact on someone isn't related to who invented the idea. It's nice to come up with new ideas—I am proud of mine and happy when they are repeated (p. 561)—but it is the exposition, the turn of phrase, the accidental time-and-place where someone heard it for the seventh time but *this* time they were prepared for it to sink deep into their psyche, and you were there, doing it in a way they were happy to accept. This is truly having positive impact on others, "changing the world" for the better, one person at a time. I have received thousands of messages over the years, saying that something I wrote changed them, or was "what I really needed to hear today." More even than a "good use of time," it might be the definition of life's purpose.

---

* *Editor's Note in 2025:* This was written in 2010, just before I founded WP Engine. That company would go on to be a hyper-growth unicorn, which certainly answers this objection. Still, it's more instructive to see how I answered it before I earned that additional credibility.

Don't get discouraged if you're not happy with all your answers. That's a good sign—it means you're being honest about the exercise and you're not yet satisfied. Keep it in the back of your mind and look for answers while you forge ahead. Discuss the hard ones with other people to generate better ideas.

This is all just another way of being introspective.

Chapter 147:
Put down the compiler until you learn why they're not buying

New companies rarely have a problem writing code.

The problem they have is: **We don't have enough sales.**

Some actual quotes (*sound familiar?*):

> "We have 300 downloads and no sales."
> "People tell me I have a great idea, but none of them bought my software."
> "My sales/download conversion ratio is 1%. It should be 8%."
> "Folks are signing up for an account but they don't come back."

Of course everyone wants "more sales," but I'm specifically talking about the early stage of your company, when your v1.0 is shaky but has enough features that it should be more viable than it is. When your website copy is good enough that people are willing to sign up or download, but the sales aren't coming in like they ought.

credit[1407]

"Things went from bad to worse, but we're hopeful now that we're doing badly again."

This problem is solved only one way: **You need feedback from lost sales.**

Empirical data, not your own ideas about why people might not be buying.

You need to talk with the people who were interested enough to find your website, read your marketing copy, download your product, and then *give up without even an email*. That's the low-hanging fruit; those are the people who are *in your grasp*, who should be buying *today*, but aren't.

They've self-identified as your ICP (p. 317), yet your product didn't fulfill the promises you made.

As Steve Johnson[1408] says, "**All the answers are outside the building.**" (Watch his one-hour presentation on the subject at the Business of Software 2008 Video Archive.[1409])

Or as Eric Ries says,[1410] "**Not listening is the cardinal sin** ... Any other mistake can be overcome: shipping bad product, removing key features, erroneously banning community members, even kicking out a whole segment of customers."

But I find that entrepreneurs—especially technical ones—fight me on this tooth and nail. And I'm not surprised because, as usual, **I**

too used to hold the I-already-know-why, I-know-my-customers-better-than-they-do attitude.

So once and for all, I'd like to dispense with the usual arguments against getting feedback:

Existing customers are telling us to do X, so we should do X.
Customer requests are important and you *must* follow their lead, especially in the beginning. But what about the 98% of trial users who *didn't buy*? It is *they* who hold the keys to more sales! Existing customers bought *in spite of* barriers to sale, so they're no help in identifying the barriers. Listen to them to increase your product's value, but listening to them to increase sales is classic survivor bias (p. 449).

What we need is New Feature X, then people will buy.
This is almost never true. The world is filled with successful v1.0 products that lacked obvious features; in fact I challenge you to find an exception. Ben Yoskovitz wrote a great post about this fallacy[1411] (with 27 concurring comments). Even Nintendo says[1412] "the most important feature is the one no one asks for."

We need to clean up the software before we can get real feedback.
At Smart Bear, the first incarnation of our code review product was so hard to decipher, I can't understand how we got customers. They used it *in spite* of the problems, not *because* of them. If you're solving a genuine pain, people will try the software, complain about it, ask for features, and generally be engaged; if that's not happening, you're not solving the right problem or not making that obvious, and *that* is critical to getting revenue.

Have you ever worked on a software project for many years and then lived through a face-lift? After you're used to the new look, you're just embarrassed when you see the old version. It's the natural order of things. Polish isn't important if you don't have enough revenue.

I'm a user myself, so I know what's missing.
That's great, but all that means is that you have 100 ideas for new features, but "more features" is almost certainly not the problem. It means is you have a "vision" which is almost certainly not how your company is going to unfold (p. 193).

Often the real impediment to sales is as mundane as "New users are presented with a blank screen, so they don't know what to do next, so they abandon the trial," or "The installer doesn't work properly under Vista, so people give up." The fact that you're a user yourself is the *worst* position for you to be in because you can't be objective about the new user experience, and you can't put yourself in the shoes of a user possessing below-average intelligence. Which half of them possess.

There, I said it. Most of your users are dumb; almost all are dumber than you are. You are not your typical user.

Apple just knows what's cool. So do we.
This is a common misconception, easy to believe because Apple does keep product development close to the vest. However, it's completely untrue. Steve Jobs specifically talks about [1413] getting feedback from customers.

We can't afford to delay the v#.# release.
If you have no real evidence that revenue will suddenly improve with the next release, why do you think it's important to release it? Just because it has "more stuff?" The only reason to be excited is because it's different, and since the status quo isn't working, you've *got* to try something different. But is that "stuff" why people are downloading but then abandoning? Until you can answer that question with empirical data, there's no reason to believe the new stuff will be more compelling than the last stuff.

Getting revenue is a marketing/sales function; I need to be heads-down in the code.
In a startup, it's *everyone's* job to get revenue. Sure, the usual day-to-day activities should be divvied up between founders; not everyone needs to write letters to bloggers and be glued to Twitter live-search. But if you don't know why people aren't buying, that's the #1 bug and the #1 feature you need to be working on. There's lots of ways (see below) to change the product or website *in under a day* that will begin fixing the problem. Saying "it's marketing's job" really means "I'm not going to help get revenue." Unacceptable.

Hopefully by now you're convinced to get more feedback from lost sales, but how do you go about doing it?

Here I've posted eleven specific ways to get more feedback,[1414] almost all of which take less than a day to implement.

And here's I've described my system for creating questions and conducting customer interviews (p. 239) that I've used to build two unicorns.

So you have no excuse.

Chapter 148:
Sacrifice your health for your startup

"Pencil."

Editor's Note: This was written in 2009, before "Hustle Culture" and indeed two years before Lean Startup. It was somewhat controversial even then, but did not have the stigma of modern ideas of work and life, and especially what some call toxic hustle culture. I stand by the assertion in the original article—that startups take all the time, and take sacrifice, without also accepting the ways people interpret these things more than 15 years later.

The Internet is full of good advice about how to lead a healthy, balanced work/home life.

- Leo Bauboa of Zen Habits built his Technorati 100 blog on one hour a day,[1416] leaving plenty of time for a day job and a family.
- Tim Brownson reshuffles our priorities[1417] so we realize what's important to accomplish and what's not important to worry about.
- Merlin Mann of 43folders shows us how merely admitting what we don't like[1418] about ourselves and our life leads to a vast menu of options for fixing it.
- Penelope Trunk demonstrates that the point of a job is fulfillment and happiness,[1419] not the blind pursuit of money.

If you don't have your health and your family, nothing else matters. On your deathbed will you wish you had worked longer hours or been a better parent? Will you wish you had spent more time Twittering or more time exercising, extending your life by five years?

Compelling. And yet, in my experience this attitude is not the path to success in small business.

Maximizing your chance for success means sacrificing health and family.

This sounds controversial, but it's not just me:

- Jeremiah Owyang of Web Strategist:[1420] "How do I Keep Up?" This is one of the most common questions I get from folks, or a variant: "Do you sleep?" or "Do you have a family?" I can answer succinctly: "I don't, in shifts, and yes... I think." ... I'm lucky I fell into my passion. It comes with costs however, I'm out of shape, stressed, I don't sleep well, and my blood pressure is up.
- Mark Cuban, self-made millionaire and owner of the Dallas Mavericks on how he achieved success:[1421] "I slept on the couch or floor ... Because I was living on happy hour food, and the 2 beer cover charge, I was gaining weight like a pig. But I was having fun. ... Every night I would read [software manuals], no matter how late. ... I remember sitting in that little office till 10pm ... I would get so involved with learning that I would forget to eat ...
- More from Mark in an interview with YoungMoney Magazine: Question: "Did you have to sacrifice your personal life in order to become a business success?" Answer: "Sure, ask about five of my former girlfriends that question. I went seven years without a vacation. I didn't even read a fiction book in that time. I was focused."
- Penelope Trunk (yes, she has insights on both sides of this issue) on how all-consuming her company is:[1422] "I'm desperate. ... You're always sick, but not take-a-day-off-work sick. ... So I suffer with the pink eye, because it's not having all that gross green discharge yet, so I think I can deal with it after funding. ... I diagnose my [temporary] blindness as stress related. ... I say, 'My eyes are nothing compared to the pain of raising money.' ... There's no time for family.

credit 1423

"So what," you could argue, "just because many successful entrepreneurs are workaholics doesn't mean that's the *only* path to success."

Indeed, study after study has shown that "working more hours" doesn't translate into "accomplishing more shit." If you're not getting enough sleep, for instance, working extra hours doesn't make up for your foggy brain.

Also, optimizing *how* you spend your time can increase productivity several times over (p. 925)—an increase you couldn't possibly match by working more hours.

Yeah, but here's the problem.

The "Rule of Closets" is that the amount of crap you own will expand to fill all available closet space. You can create more space by adding shelves and organizers, but then you'll soon discover you have more stuff.

Well I have a "Rule of Time in Startups": **How much time does a bootstrapped company take? All of it.**

Even ten people could hardly keep up with everything you do in small business—creating, consulting, designing, fixing, self-promotion,

blogging, networking, bookkeeping, taxes, customer support and cultivation, reading startup blogs for ideas and inspiration (!), and all those little crappy things like losing an afternoon troubleshooting your fancy outsourced IP phone system that was supposed to let you "work from anywhere."

One, two, or even three people can't do everything, so *of course* it takes *all* your time. If you're working a day job while starting something on the side, *of course* you don't have time to exercise or play with your kids before bed.

It takes obsession to make a little company go. Forget "passion"—everyone's favorite word—it's "obsession." It's not just that you love working, it's that you can't stop working. You're putting your entire self on the line—your finances, your career, your ideas.

The obsession is there even when you're away from the office, having lunch with a friend or reading to your kids. As my wife would frequently point out in the early years of Smart Bear, my "mental and emotional bandwidth" was entirely consumed. You're physically there, but you're not really there.

Read those quotes above again and you'll see not just passion but **self-destructive devotion**. You don't put yourself through this meat grinder just because you "like something a lot."

"If you love it so much, why don't you marry it?"

Exactly.

Of course those life-coaches are still correct: This isn't a great way to live your entire life. You need to accept that this is going to happen and ask whether it's OK to incur this penalty right now. For me, I did all this in my 20's when I had no kids, I had enough savings to risk everything for a while, and I had a wife who had her own business and who therefore understood how much work it took and why I was spacing out over dinner.

Bottom line: **Every *successful* bootstrapper I know puts work before self.** (Until financial freedom is achieved.) I did too.

Chapter 149:
Letters to Joel Spolsky

Dear Joel,

I regret to inform you that I must decline your invitation to be a featured guest blogger for Joel On Software.[1424]

I realize this will come as a shock, especially given my well-documented need for attention.

The fact is, I don't care how many thousands of readers you have, how many millions of dollars of software you sell, or how many minor celebrities worship you. At the end of the day, you appear in a little window in an RSS reader. You fill in a template consisting of a cute story tenuously connected to

a dramatic point, inspiring wanna-bes to commiserate and laugh with indignation at the stupidity of others.

While they've been laughing, I've wondering whether you practice what you preach. You admonish programmers who don't understand Unicode,[1425] yet five years later our copy of Fogbugz still cannot receive email from Korea because of a character encoding issue.

Also, are you out of gas? Your column in Inc Magazine consists of 1300-word reproductions of chapters from your book which themselves are reproductions of blog entries you wrote in 2001. And your blog has turned into announcements for products and trade shows.

I can already hear your fanboys calling for my head, but from where I'm sitting, you're a celebrity who is cashing in on fame, no longer compelled to have new ideas.

But introspection isn't your thing. Admitting you've been wrong or that you don't take your own advice would crack your well-crafted façade.

I'm not like that, and I can't pretend otherwise for you or your readers. I'm afraid the answer is no.

Sincerely,
Jason

Dear Mr. Spolsky,

I'm not sure if you received the last email I sent. I hope not. I used Outlook's "recall this message" feature, but sometimes that doesn't work. (That's Microsoft for ya, am I right? Ha ha!)

Anyway, I'd like to apologize for the things I wrote. I feel I've done both of us a disservice by refusing your generous offer to be a featured guest writer for Joel On Software.

If you want to know the truth, my unwarranted outburst stems from a core insecurity. Had you rejected my article, I would have been crushed. I guess this was my way of rejecting you before you could reject me. Juvenile, I know.

In fact I have deep respect for what you've done for the software development community over the past decade. I myself have been inspired by you since 2000; I can still remember the glee of getting new articles delivered to my inbox.

As an entrepreneur, you've taught me everything from how to hire great people,[1426] how to think properly about bootstrapping,[1427] how new projects help you cope with burn-out,[1428] and even how to run tech support.[1429] In fact, there's very little I do each day that isn't influenced by you in some way.

That's incredible, if you think about it.

What I'm trying to say is, I would be honored to accept your invitation, and I trust that you will disregard my first email.

Apologetically and humbly yours,
Jason Cohen

Hi Joel!

I haven't heard from you, so I'm forwarding a copy of an email I sent earlier this week.
You must get an ass-ton of email! So no hard feelings.

Talk to you soon,
Jason

Hi Joel,

Oh man, that article about hanging the blinds at Fogcreek[1430] was awesome. Did you really do all that? Of course you did, it was in the photo! I loved how you tied in the army story—it's really motivational.

I'm so glad Inc is featuring you. They need someone to speak truth to power and put the stuffed suits in their place. You're like the Moses of software developers! What's next, the New York Times? Why not!

Speaking of articles, I've got some article ideas I'd love to discuss! I know you're super-busy—that's what I keep telling my friends. They're such nervous nellies—they think you're ignoring me! A quick little two-second reply from you would really reassure them. Thanks!

+1 for Joel in the NYT!

Jason

Joel-

Quick idea: I was thinking of doing an interview series about how your writing has inspired successful software projects. Maybe even make a short film? You could attach it to your next "Interning at Fogcreek" DVD. What do you think?

Here's what I'd say: Your three[1431] part[1432] series[1433] on designing software for real people permanently changed my perspective and continues to be my bible. It's the kind of thing you have to re-read every few months to make sure you're building great, usable software.

P.S. I still haven't heard back about the guest post. Should I be worried?

Thanks again,
J

Hey hey J-Spol!

I was just telling a friend about your offer. You know, all I have to say is "Joel" and everyone knows exactly who I'm talking about. I guess that's how you know you've made it!

Anyway, this friend thinks that if you were truly interested, we would have had more conversations by now. Imagine how surprised she'll be when you publish my article! Ha ha, we'll both get a kick out of that.

Let me know.

Waiting expectantly,
Your boy JC

Hi Joel,

This will be my final email. I don't want to seem like a stalker!

So it turns out I have some influence over one of your interns (one of those friend-of-a-friend-who-owes-a-favor-to-a-friend type deals). He (or she!) set up me with a Copilot [1434] account behind the FogCreek firewall, so I've been playing with the Joel On Software CMS myself.

Seems like it's a custom job. No problem—I'm Smart and I Get Things Done [1435]—I'll figure it out.

So you should see my article appear soon! I'm glad I found a way we could work together without interfering with your schedule. Cheers!

—Jason

Shamelessly modeled after "Dear Oprah" from Steve Almond's[1436] *fantastic short story book Rants, Exploits, and Obsessions (Not that you asked).*[1437] *Good artists copy (p. 977); great artists steal. (Said by Steve Jobs, stealing a quote from Pablo Picasso.)*

Chapter 150:
How to get customers who love you even when you screw up

"I'd like a drunken college kid right here."

credit 1438

During the first year of Smart Bear's existence, my software was crap. How did I get customers, and why were they so vehemently loyal to what was clearly a wobbly, new product from a teeny tiny company-of-one?

Because of people like Tom.

So Tom calls up one day...

Now wait, understand this is already a newsworthy event! Remember we sell software to software developers, legendary for their phone-aversion. (I'm no exception!) So let me try that again:

Tom *called* me. On the *phone*.

Tom wants to talk about new features. What a relief—for six weeks it's been nothing but bug reports. Real bugs, I admit. In fact, Tom had single-handedly debugged a significant amount my shitty code, even enlisting his own employees for the cause. (Why had he done that?)

Tom lists 20 new features he'd like to see. When does he expect delivery? "Oh, I know you're just a one-man shop, so just do your best. If you get through this half as fast as you get through bugs, we'll be fine."

Whozajigga-wha? I never said I was a one-man shop!

"We" always use the first-person-plural when talking about "our software" and "our release cycle" and "our tech support." My website was professional-looking (p. 853) (uhhh right?). Tech support always came from `support@smartbear.com`; my name was never on it.

So was Tom the Sherlock Holmes of small business façades? Hardly. The web site doesn't look *that* professional. Tom's gotten sales support, tech support, and bug fixes for weeks now; he recognizes the same style and phrases. He always called the main line and never found anyone but me.

Duh!

But this was going to be a problem (or so I thought). See, Tom worked for a big company (I don't have permission to say which) with thousands of employees and billions in revenue. Big companies don't buy software from one-man shops. Or so I've been told (p. 751).

I almost puked out the mantra of how, yes, I'm the only full-time employee, but I use consultants for stuff when the workload goes up (which wasn't true). And I almost went into defensive mode, talking about how good our/my service was and all that.[1439]

But fortunately I recognized that Tom didn't want to hear any of that. Tom was saying: "I know who you really are, and I accept it. I still want to do this. How about a few features since we put up with those bugs?"

I *had* to match that honesty. Anything else would be an insult to his intelligence and a step backward in the relationship.

> *"Truth is such a rare thing, it is delightful to tell it."*
>
> *–Emily Dickinson*

It wasn't until I visited him in Ottawa that I fully understood why Tom was so solicitous. We met with two of Tom's bosses in a small office to discuss widespread purchase and roll-out of our peer code review tool.

Tom introduced me in a way I didn't expect: "Half a year ago I found this company in Austin. They had the beginnings of a code review tool. I've been guiding their development so now it works perfectly for our environment."

Hmmm, that's not exactly accurate... or is it? I'm on the spot, so I just nod in agreement.

The bosses questioned the utility of the tool. How much time could it save? Tom had the answer: "I do sixty code reviews every day."

He might as well have said "I can squeeze crude oil from cow patties." One boss flatly said "That's impossible." Honestly I'm not sure whether he was referring to Tom's fortitude or the tool's efficiency. But they looked at Tom's evidence, and approved the roll-out. I got a big order, thanks to Tom. But again I ask you, why?

In that moment I understood Tom's motivation: **Tom was a hero.**

Tom had figured out how to deliver code with fewer bugs and was training his new hires faster than other team-leads. Tom didn't do this by paying IBM or implementing a process he read about in Dr. Dobbs—he found a little company (us... I mean "me"), and he was now *personally* responsible for directing our product development. We jump when he says jump, therefore the perfect product (for their company) had been forged.

All due to his prescience, product development prowess, and a relationship he had forged with the founder.

Don't forget, this was before "relationship" became the buzzword of modern marketing—before blogs and Twitter and back when Facebook wasn't just for boomers and neighborhood groups.

I can't begin to tell you the amount crap Tom put up with over the years. We're good at this now (no really, 15 people counts as "we!"), but back then screens would lock up, reviews would inexplicably disappear, installers would install the wrong files, and occasionally we'd run computers out of memory.

He put up with all of it why? Because it was just him and me. Because he knew I always kept my word. Because he knew he could stick

his neck out for Smart Bear and I wouldn't let him down. Because he knew I would ensure that as the product changed it continued to solve his problems better, because I didn't want to let him down.

So he pinned his own reputation on it and won. As a bonus, he lived vicariously through Smart Bear as a product designer.

If I hadn't fessed up and behaved honestly, perhaps none of this would have happened.

What will your first hundred customers look like? Big, established companies with bureaucratic purchasing systems that you will bluff your way through? Well-known consumer-advocacy bloggers?

No, they'll be early-adopters—folks who like trying new stuff and like working with new companies who still have spark and something to prove. Folks who want to be part of the creative process and be able to tell their friends that they were there at the beginning.

If you pretend to be something you're not, they'll see right through it. Then what have you done? **You've lied to those who would have loved you for who you are** **(p. 317)**; that's not how you build a relationship.

It doesn't mean never telling a lie. Authenticity (p. 939) doesn't mean abandoning social white lies. We all know the difference between outright lies and the business equivalent of "No those pants don't make you look fat."

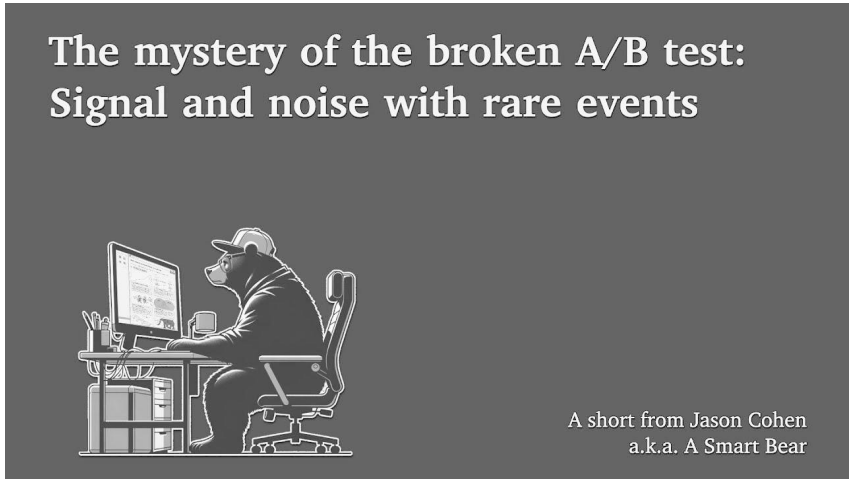
Theoretically, honesty should be easier than dishonesty. After all, "if you tell the truth you don't have to remember anything" (Mark Twain). It's true that in business you're so accustomed to fluffing your feathers and putting on a show, it can be hard to remember to act like a normal human being.

"Be yourself" (p. 1433) is just as hard in business as it is in personal life. But it's worth it.

And it's one of the few advantages small companies have (p. 295) over big ones. Use it.

Chapter 151:
Easy statistics for A/B testing and hamsters

SIGNIFICANCE · DERIVATION



Watch on YouTube[1440]

This video explains the concept, as well as the statistical principle that explains the mental fallacy that tricks us when we reason about rare events.

So you've got your AdWords test all set up: Will people go for the headline "Code Review Tools" or "Tools for Code Review?"

Gee they're both so exciting! Who could choose! I know, I know, settle down. Welcome to A/B testing.

Anyway, the next day you have this result:

	Variant A "Code Review Tools"	**Variant B "Tools for Code Review"**
Clicks:	31	19

Is this conclusive? Has A won? Or should you let the test run longer? Or should you try completely different text?

The answer matters. If you wait too long between tests, you're wasting time. If you don't wait long enough for *statistically conclusive* results, you might *think* a variant is better and use that false assumption to create a new variant, and so forth, all on a wild goose chase! That's not just a waste of time, it also prevents you from doing the *correct* thing, which is to come up with a *completely new test*.

Normally a formal statistical treatment would be too difficult, but I'm here to rescue you with a statistically sound yet incredibly simple formula that determines whether your A/B test results really are significant.

I'll get to it in a minute, but I can't help but include a more entertaining example than AdWords. Meet Hammy the Hamster, the probably-biased-but-incredibly-lovable tester of organic produce:

Watch Hammy the hamster on YouTube[1441]

In the movie, Hammy chooses the organic produce **8 times** and the conventional **4 times**. This is an A/B test, just like with AdWords... but healthier.

If you're like me, you probably think "organic" is the clear-cut winner—after all Hammy chose it *twice as often* as conventional veggies. But, as so often happens with probability and statistics, **you'd be wrong**.

That's because human beings are notoriously bad at guessing these things from gut feel. Most people are more afraid of dying in a plane crash than a car crash, even though the latter is 1000x more likely.[1442] On the other hand, we're amazed when CNN "calls the election" for a governor with a mere 1% of the state ballots reporting in. We also can't distinguish between patterns and noise (p. 1487).

Okay okay, we suck at math. So what's the answer? Here's the bit you've been waiting for:*

* For the mathematical derivation, see the end of the article.

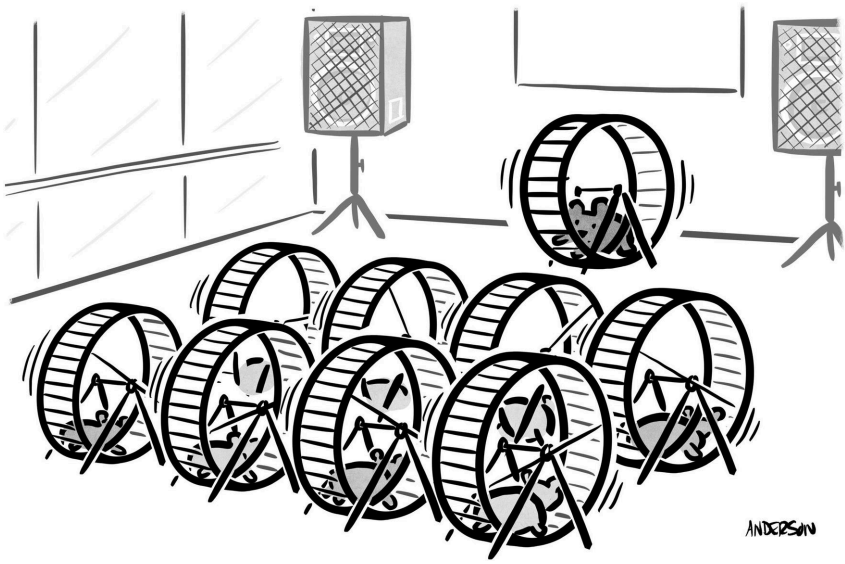
DETERMINING WHETHER AN A/B TEST IS STATISTICALLY SIGNIFICANT

1. Define N as "the number of trials."
   For Hammy, $N = 8 + 4 = \mathbf{12}$
   For AdWords, $N = 31 + 19 = \mathbf{50}$
2. Define D as "half the difference between the 'winner' and the 'loser.'"
   For Hammy, $D = \dfrac{8 - 4}{2} = \mathbf{2}$

   For AdWords, $D = \dfrac{31 - 19}{2} = \mathbf{6}$
3. The test result is statistically significant only if $D^2 > N$.
   For Hammy, $D^2 = 4$, which is *not* bigger than 12, so it is *not significant*.
   For AdWords, $D^2 = 36$, which is *not* bigger than 50, so it is *not significant*.

So your AdWords test isn't statistically significant yet. But you let the test continue to run. The next day you find 31 more clicks for variant A, and 19 more clicks for B. Rerunning the test, the measured difference is now significant:

	Variant A	**Variant B**	N	D	D^2	**Stat Sig?**
Day one:	31	19	50	6	36	No
Day two:	62	38	100	12	144	Yes



credit [1443]

A lot of times, though, you keep running the test and it's still not significant. That's when you realize you're not learning anything new; the variants you picked are not meaningfully different for your readers. That means it's time to come up with something new.

When you start applying the formula to real-world examples, you'll notice that **when N is small, it is difficult—or even impossible—to be statistically significant**. For example, say you've got one ad with 6 clicks and the other with 1. That's $N = 7; D = 2.5; D^2 = 6.25$. So the test is still inconclusive, even though A is beating B six-to-one. Trust the math here—with only a few data points, you really don't know anything yet.

The smaller the N, the bigger the difference needs to be, to be detectable. Specifically, results are significant only if the ratio between A and B is larger than $\frac{\sqrt{N}+2}{\sqrt{N}-2}$. So for example, if $N = 50$, as it was in our "day one A/B test" example, the winning variant needs to be almost double the number of clicks as the losing variant in order to have a detectable difference, e.g. a conversion rate of 10% versus 5%.

This is a huge difference; it's great if you find something so dramatically better, but this is rare, and therefore you can almost never find a significant A/B test given only 50 clicks to analyze.

When $N = 100$, the winner needs to be at least 50% higher than the loser (which it was by the second day in our example). It takes $N = 1800$ to detect the case where the winner is only 10% larger than the loser.

And this is bad news for A/B tests, because often one variant isn't better than the other by more than 10%, e.g. a "2.4% conversion rate" versus a "2.2% conversion rate." What does this mean, especially if you don't have large N? It means **you need to be seeking big differences**, not subtle ones. Test wildly different designs, rather than tweaks. Tweaks can only be tested when N is enormous.

I hope this formula will help you make the right choices when running A/B tests. It's simple enough that you have no excuse not to apply it! Human intuition sucks when it comes to these things, and A/B testing tools often use misleading or incorrect math, so let this formula help you draw the right conclusions.

APPENDIX FOR THE MATHEMATICALLY INCLINED: THE DERIVATION

The null-hypothesis[1444] is that the results of the A/B test are due to chance alone. The statistical test we need is Pearson's chi-squared.[1445]*

* Not the Student t-test as is commonly claimed by people online who have only a passing familiarity with statistics; the t-test is appropriate with continuous, normally-distributed random variables, whereas χ^2 is appropriate for categorical random variables from independent trials and arbitrary probability distributions, which is what an A/B test is.

The definition of the χ^2 statistic follows, where:

m = number of possible outcomes;

O_k = observed quantity of results in category k;

E_k = expected quantity of results in category k:

$$\chi^2 = \sum_{k=1}^{m} \frac{(O_k - E_k)^2}{E_k}$$

In the simple case of a two-variant A/B test, $m = 2$. O_1 and O_2 are the observed results, and definitionally $N = O_1 + O_2$. The expected result under the null-hypothesis is that the quantities fall equally into each category, therefore $E_1 = E_2 = N/2$.

Plugging this into the definition:

$$\chi^2 = \frac{(O_1 - \frac{N}{2})^2}{\frac{N}{2}} + \frac{(O_2 - \frac{N}{2})^2}{\frac{N}{2}}$$

The first numerator can be rewritten in terms of O_1 and O_2 by substituting $N = O_1 + O_2$, and this results in our variable D^2 as defined in the main text:

$$\begin{aligned}
\left(O_1 - \frac{N}{2}\right)^2 &= \left(O_1 - \frac{O_1 + O_2}{2}\right)^2 \\
&= \left(\frac{2O_1 - O_1 - O_2}{2}\right)^2 \\
&= \left(\frac{O_1 - O_2}{2}\right)^2 \\
&= D^2
\end{aligned}$$

We can repeat with the second numerator, and so the expression simplifies:

$$\begin{aligned}\chi^2 &= \frac{(O_1 - \frac{N}{2})^2}{\frac{N}{2}} + \frac{(O_2 - \frac{N}{2})^2}{\frac{N}{2}} \\ &= \frac{D^2}{\frac{N}{2}} + \frac{D^2}{\frac{N}{2}} \\ &= \frac{2}{N}(2D^2) \\ &= \frac{4D^2}{N}\end{aligned}$$

Now that we have a simple formula for the chi-squared statistic, we refer to the chi-squared distribution to determine statistical significance. Specifically: What is the probability this result would have happened by chance alone?

Looking at the distribution[1446] at 1 degree of freedom, we must exceed 3.8 for 95% confidence and 6.6 for 99% confidence. For this simplified rule-of-thumb formula, I selected 4 as the critical threshold. Solving for D^2 completes the derivation:

$$\chi^2 > 4$$

$$\frac{4D^2}{N} > 4$$

$$D^2 > N$$

□

(And if D^2 is more than double N, you're well past the 99% confidence level.)

Deriving the other statement in the article—that the ratio between the two variants needs to exceed a certain threshold to be significant—start with the boundary condition of being significant, and derive the values of A and B in that case:

$$
\begin{aligned}
D^2 &= N \\
D &= \sqrt{N} \\
A - \frac{N}{2} &= \sqrt{N} \\
A &= \frac{N}{2} + \sqrt{N}
\end{aligned}
$$

Similarly, given that $B = N - A$:

$$
B = \frac{N}{2} - \sqrt{N}
$$

And so:

$$
\begin{aligned}
\frac{A}{B} &= \frac{\frac{N}{2} + \sqrt{N}}{\frac{N}{2} - \sqrt{N}} \\
&= \frac{N + 2\sqrt{N}}{N - 2\sqrt{N}} \\
&= \frac{\sqrt{N} + 2}{\sqrt{N} - 2}
\end{aligned}
$$

Chapter 152:
Your idea sucks, now go do it anyway

FACE-PALM-PILOT · NEVER-ENDING STORY
NOT-SO-SMART BEAR

"My idea isn't good enough yet" explained a friend who is thinking of starting his own company. He's waiting for the idea to be completely fleshed out before taking the leap.

Newsflash: **Your idea probably sucks, and it doesn't matter** because your business will probably turn out to be something completely different.

Sounds wrong? Let's see.

FACE-PALM-PILOT

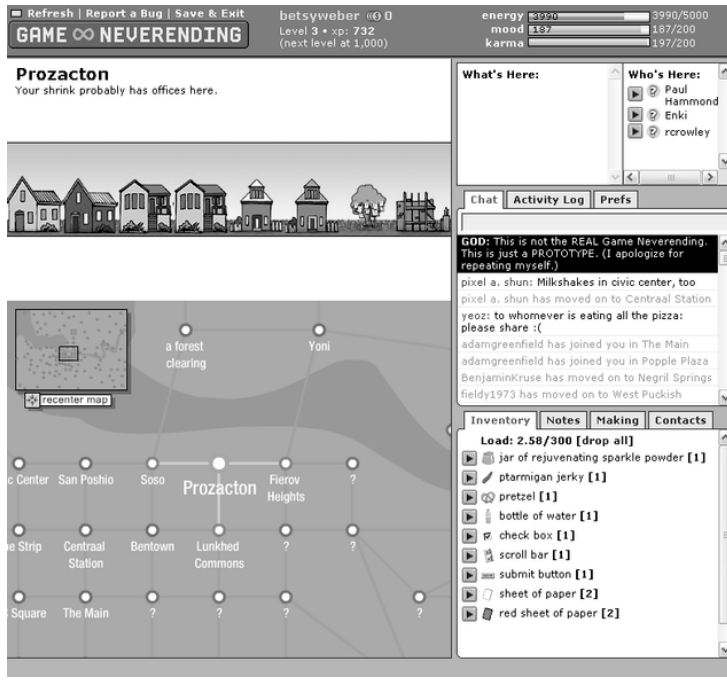
In 1998, a company received $4.8 million in funding to "beam money between Palm Pilots." I'll code-name this product: Money-Beamer.

Here's the pitch. Alice wants to give Bob some money, but Alice doesn't have cash or her checkbook. There's no ATM around. Both Alice and Bob do own palm pilots and they both previously installed MoneyBeamer and, despite having forgotten all their normal modes of money transfer, they did remember to bring their palm pilots. MoneyBeamer will allow Alice to send money to Bob. Well actually it won't, but it will *remember* that Alice *wants* to send Bob money, and once Alice gets back home and connects her Palm Pilot with her computer, and after she dials up to the Internet, MoneyBeamer will contact a server and transfer the money, provided of course that Alice has the money and didn't secretly change her mind in the meantime.

Would you have invested in them? Not with an idea like that. You'd be wrong though—it was PayPal. Their work with encryption combined with an idea for a consumer-targeted on-line banking system made it the easiest way to send money... by email. They were sold to eBay for $1.3 billion. Today they process $2,000 of payments every second.

NEVER-ENDING STORY

I'm sure you won't recognize this web-based sensation (Figure 1).



credit 1447

Figure 1: Game Neverending: An in-browser multi-player on-line game "with no way to win, nor any definition of success." (Like life!)

This is Game Neverending: An in-browser multi-player on-line game "with no way to win, nor any definition of success." (Sounds like a lot of Web 2.0 companies to me.) It never saw the light of day.

What was most interesting (to its alpha testers) was that people could share game objects by dragging them into chat windows. They saw this as a useful enhancement to chat applications in general, so as plans for the game fizzled out the engineers created a Flash application for real-time chat plus file-sharing with a particular emphasis on image-sharing.

Unfortunately the Flash application was *only* real-time—your pictures didn't stick around when you closed it. And this was fatal because it turns out people were interested in the *sharing* part more than the *real-time* part. So in yet another upheaval they rewrote the Flash appli-

cation as a regular website and lo, Flickr was born. Now it's the largest photo-sharing site in the world with 3 billion photos and 5,000 more uploaded every minute.

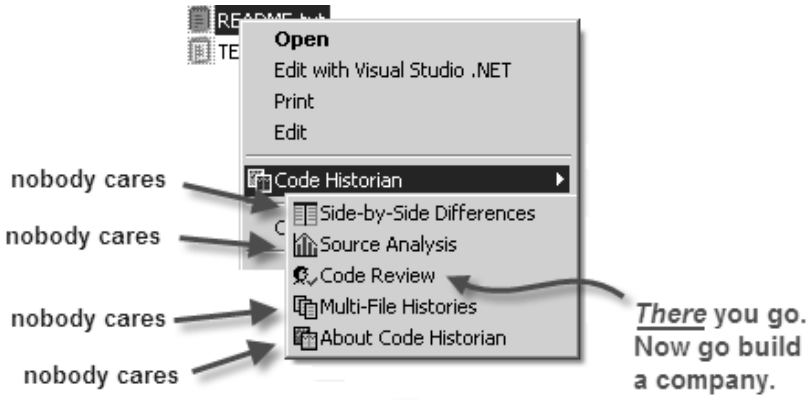
NOT-SO-SMART BEAR

Of course a rant like this wouldn't be complete without self-deprecation, so let's accompany the Ghost of Christmas Past into the annals of my own company, Smart Bear.[1448] My first idea was a product called Code Historian; it could dig through the history of a file and show you what changed. Accurate name, but turns out to be almost useless.

Like an adolescent,[1449] the company went through many embarrassing stages (forgive the broken images, 'tis the way of the Way Back Machine[1450]):

1. Mar 24, 2003:[1451] Hideous. "Do one thing and do it poorly."
2. Dec 22, 2003:[1452] Fugly. "Three products... is that enough for a Suite?"
3. Oct 10, 2004:[1453] Lame. "Everything above the fold, most expensive first."
4. Jan 11, 2006:[1454] Getting there. "You really need a graphic designer. No, really."
5. Sep 10, 2007:[1455] Ain't bad. "At least you admit 'code review' is all that matters."
6. Present day:[1456]* Ummm... "Where did those other products go?"

* Editor's note: This was written in 2008.



At one point we were selling six different tools; the only one that mattered in the end was Code Reviewer. Perhaps a screenshot will make this clear:

The point of all this isn't to berate anyone for their crappy ideas. In fact, just the opposite—the point is that **it doesn't matter what your first idea is**. First, it's probably wrong. Second, the only way to find the right one is to try the wrong one and see what happens. You won't find it by fiddling around with PowerPoint slides and Photoshop mock-ups.

So get out there and make some mistakes! As Neil Davidson said:[1457]

> You don't need stratospheric growth and a billion-dollar addressable market to bootstrap a software company. A $50,000 market opportunity is enough to get you off the ground—once you get started you'll figure out the rest.

(Neil is the co-founder of Red Gate Software. It started as yet-another-online-bug-tracking-system that no one cared about but is now a popular purveyor of fine SQL database tools with 95,000 customers to their credit.)

Chapter 153:
Breaking the Rules

Pop quiz: Which of the following two paintings was made by Leonardo Da Vinci (a Renaissance painter from the early 1500s), and which was made by Pablo Picasso (a surrealist from the early 1900s)?

Trick question! Both are Picasso's.

Picasso didn't start out doing crazy, inventive things; he first learned the rules of standard, classical painting and proved himself a

master of that genre. A genius in fact—he completed "The First Communion" (the painting on the left) when he was just 15 years old.

There are those who look at "Dora Maar au Chat" (the painting on the right) and can't understand how it can even be called "art," much less good art, important art, certainly not worth *$95 million dollars*, which it sold for at auction in 2006.

But "The First Communion" proves something important: That Picasso painted "crazy stuff" because it was exactly what he wanted to paint, not because he wasn't able to paint "properly," and not because it was the only thing he could do. It was a choice.

The choice is everything. Had Picasso never proven he could paint classically, he could be dismissed as a hack. When you know the rules, you're allowed to break the rules.

This lesson applies all the time. Let's take a cardinal rule of advertising: "Never use negative words in headlines." You don't want people associating negative words with images of your product, not even subconsciously. It's important both for advertising and branding.

Volkswagon broke this rule with fantastic results (Figure 1).

"Lemon" is about the worst thing you can say about a car, yet there it is. Gets your attention, doesn't it? The fine print underneath is brilliant with an casual style that is as effective and relevant today as it was in the 1960s:

> This Volkswagen missed the boat.
>
> The chrome strip on the glove compartment is blemished and must be replaced. Chances are you wouldn't have noticed it; Inspector Kurt Kroner did.
>
> There are 3,389 men at our Wolfsburg factory with only one job: to inspect Volkswagens at each stage of production. (3000 Volkswagens are produced daily; there are more inspectors than cars.)

Figure 1

> Every shock absorber is tested (spot checking won't do), every windshield is scanned. VWs have been rejected for surface scratches barely visible to the eye.
>
> Final inspection is really something! VW inspectors run each car off the line onto the Funktionsprufstand (car test stand), tote up 189 check points, gun ahead to the automatic brake stand, and say "no" to one VW out of fifty.
>
> This preoccupation with detail means the VW lasts longer and requires less maintenance, by and large, than other cars. (It also means a used VW depreciates less than any other car.)
>
> We pluck the lemons; you get the plums.

It's effective only because the authors understood how to break the rule. They played with words exactly enough to get your attention but not so subtly that you missed the point. Without appreciating the rule, the balance might not have been achieved.

There's no rule that can't be broken, so long as it's broken with purpose. Rules exist to guide us—a reasonable default when we don't have a better idea—but they're not a straightjacket. In fact, the truly

innovative, inspired ideas are frequently the result of breaking a rule that no one else is willing to break.

But if you never bothered to learn the rules, you're probably just mistaken.

P.S. Picasso's full name was: Pablo Diego José Francisco de Paula Juan Nepomuceno María de los Remedios Cipriano de la Santísima Trinidad Clito Blasco y Picasso López. It was so hard, even Wikipedia got it wrong!

Chapter 154:
Limiting Options

The 1990s was the golden age of computer AI's for the board game Othello.[1458] (If there ever was a golden age...)

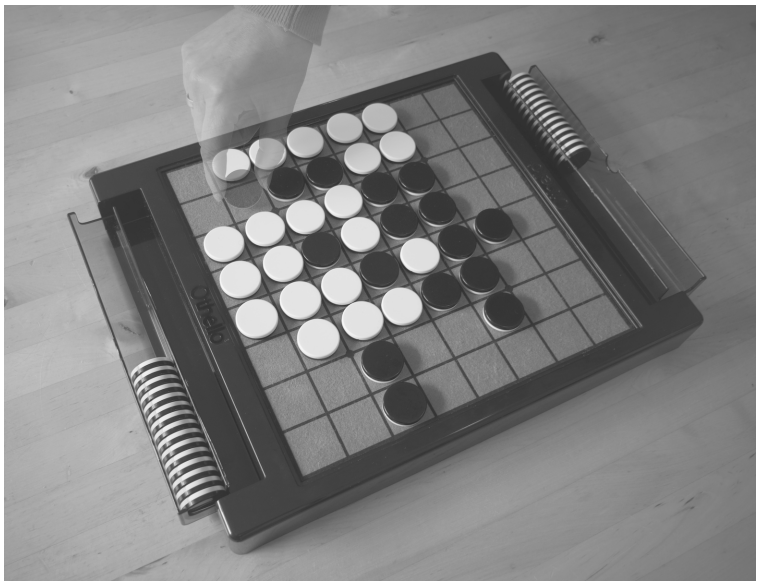
Programmers love the idea of computers beating humans at "intellectual games." For me, it's the "mad inventor" idea of creating something more intelligent than myself (whatever that means).

At the time, Checkers had been solved and Chess was close. It was Othello's turn to fall.

What's interesting is how the winning strategy worked.

The typical computer board game strategy is: Look many moves ahead and rate each resulting board position. Then pick the move that maximizes the ultimate board position you can achieve, even if the other player makes the best-possible move every time. The trick is in rating the "goodness" of a particular board position.

In Othello you win once there are no more legal moves, and if you have more tiles of your color than your opponent does. Therefore, typical metrics for "goodness" of Othello boards included things like "How many tiles of my own color do I have" (more is better), and



credit 1459

"Tiles of my color at the edge of the board are more valuable than in the middle" (the edge is a better strategic position).

What's neat is that the winning strategy used a completely different "goodness" metric. Specifically the metric was: How many valid moves does my opponent have? Fewer is better.

The flip was that it's not primarily about how many tiles you have or even the positions of your pieces. Instead it's about limiting how many choices your opponent has. **Limiting choice is more important than what the choices are.**

This principle is common in defensive theories of sports. The defense can never cover all contingencies so instead it forces the offense into higher-risk, lower-percentage moves. In basketball you can't simultaneously cover the long shot and the charge, so you elect to give up the low-percentage three-point shot. In football you can't cover both in-routes and out-routes, so you force a throw into as much traffic as possible.

In business your competitors always have options. **How can you limit their choices to things that are low-percentage or expensive?** Make them have to spend more money, get more lucky, or be more creative.

Here's some:

Honor the competitor's coupons
This eliminates the coupon as a way for your competitor to "beat" you; coupons are no longer a "choice" to develop a competitive edge.

Overpay for the best advertising slots
You are paying too much if you measure the direct ROI only, but there's also the value of forcing competitors into a low-converting ad position, leading to many fewer customers at any price.

Pay for exclusive deals with the largest affiliates
You are (again) paying too much if you measure the direct ROI only, but you force competitors to work with affiliates who can generate only 1/10th the leads.

Exclusive deals with brand-name suppliers
Often your suppliers are invisible to your customers, but sometimes their identity is known, either because it cannot be helped or it's part of your value-proposition that you're integrated with a well-known brand. If that integration is something customers value, and you can make it exclusive, then your competitors have to find other, less-desirable ways to achieve the same thing, or else completely cede that capability to you.

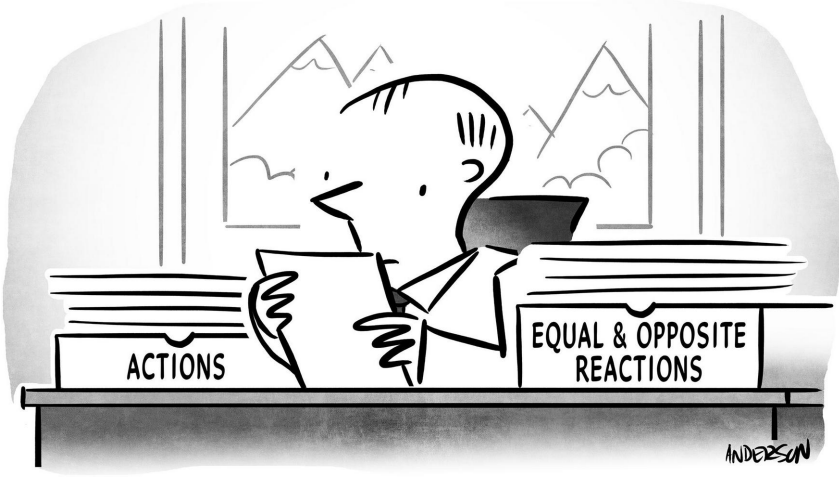
Patents (but...)
Patents limit how a competitor can compete. If no one else can use your method, they'll have to think of another way. But note: This is useless in software, where it's always easy to do it another way. Software companies don't protect themselves from competitors with patents; fortunately dozens of moats (p. 761) are potentially available.

We've done a few things at Smart Bear to limit options. For example, we're so strongly established as "the experts in code review," it would take an expensive, time-consuming effort for someone else to claim par, much less surpass what we've done. We literally wrote the

book, we did the largest-ever study on code review, we have years of history, we have the most popular tool. It's possible, but very hard, for someone else to compete on that particular basis.

This matters because in my opinion, being "the expert" is the best qualifier for our particular market, which is the Enterprise; our customers are people like Microsoft, Qualcomm, Intuit, IBM, and Adobe. Our customers are less interested in "lowest cost" or "simplest installation" (even though our installation *is* simple). If my opinion is true, I've removed the single best product-positioning choice from competition, and they'll have to find a less desirable, lower-percentage path.

Chapter 155:
The “Opposite Test”



credit 1460

I'm sick of generic feature/benefit bullet points. They're too easy to make fun of. Here's a sampling from a website that will remain nameless to protect the guilty:

- Easy to use
- Robust features
- Innovative systems
- Customer-first

Really, it's easy to use? As opposed to what, difficult and temperamental? Robust, huh? Great, because from here it looks tenuously held together, the slightest breeze threatening to crumble its delicate construction, so it's good to know that, actually, it's robust. Oh I'm sorry, the *product* isn't robust, the *features* are robust, whatever *that* means.

If you want to *not* stand out from the crowd, use statements that everyone uses. Would anyone claim to be non-innovative? Anyone claim that they put customers third?

So here's my* Opposite Test: For each feature/benefit bullet point, construct its negative and see if that statement is ridiculous. Would anyone be able to construct a rational strategy with that negative? Perhaps a competitor already has! If the negative is indeed ridiculous, if it would be impossible to have a product or positioning or strategy that included the negative, it means this bullet point is trivial, obvious, mandatory, or at least undifferentiating from the competition.

It means it's weak, it's boring, and most importantly, it's meaningless. And it's taking up space (on the page and in your brain) that should be occupied by meaningful, powerful, differentiating things, upon which you're basing your product strategy.

---

* Although I'm proud to have come up with this independently, this is a well-worn idea with many famous proponents, such as Roger L. Martin ("Is the opposite stupid on its face?"[1461]) and Al Ries and Jack Trout in *Positioning: The Battle for Your Mind* ("Anytime you come up with a positioning idea, test it out by asking yourself this question: If a competitor did the exact opposite, would it make sense? If it doesn't, then you're saying nothing.") It's also sometimes called the Reversibility Test. This convergent thinking only lends more credibility to the idea.

Let's apply the test. The negative of "Easy to use" is "Difficult to use." That would be a pretty funny statement! No one would ever claim that, so throw it out.

The negative of "Enables communication" is "Blocks communication." Crazy; no one would admit their tool does that.

The negative of "Stores files as big as 100 terabytes" is "Cannot handle huge files." Not ridiculous, in fact this is sadly true of many systems. It passes the test.

The negative of "Fully open source" is "Closed source." Of course that *is* the strategy used by most companies, so it passes the test.

"Fully backward-compatible, even after twenty years." This passes the test. Most software introduces breaking changes at some point, to enable new architecture or new features. In fact this is an important strategic decision. Backward-compatibility is important when you have millions of users with on-premise software where some components are 10 years old and no longer updated (like plugins in WordPress), and therefore compatibility is a feature. But it's bad in that it hamstrings designers on UX innovation, product managers on workflow and feature innovation, and engineers on architecture and performance innovation; sometimes a breaking change is required to stay relevant and modern.

Here's a good one from our own product[1462] at Smart Bear: "Integrates with seven version control tools." Negative: "Not integrated with version control" or "Integrated with [one tool]." Not particularly funny; in fact each of these statements *are* true of all our competitors. And it can be a good strategy to be deep and feature-rich against a single API, rather than support a wide number of APIs, limited to features that are common to all. So this statement differentiates ourselves in a specific way, and the opposite is a valid—and actually-practiced—product strategy.

Another good way of understanding whether you're passing the Opposite Test is **whether there are negative consequences** from your statement, which you acknowledge. Continuing the examples above: The advantage of supporting multiple version control systems is com-

patibility, but the advantage of supporting only one is a deeper, more sophisticated workflow integration. And since a single team probably uses only one version control system, a single team might prefer the "one version control" system over the "multi." Thus, selecting "multi" anyway has a direct negative consequence, and therefore it is a real decision.

If you're using generic bullet points now, you'll find that replacing them isn't easy! You have to really think about what's strongest about your product, about how specifically it beats the alternatives, and how make it pithy. This is a useful exercise in itself.

One exception to the Opposite Test: You can use a generic if it's your single biggest differentiator, where you're truly 10x better than the competition along that dimension, and therefore you really "own" this concept as your identity and value.

A good example here is "Fastest." The negative is funny ("Slow operation means lots of time staring into stagnant progress bars"[1463]). But if you make it your highest priority, it can work. Make your bi-line "The fastest ." Prove it with benchmarks. Explain how speed is not only about saving operator time (the obvious benefit) but how it enables entirely new features. For example, perhaps operation X is typically so slow that people can't take advantage of it. But since your system can complete operation X in two milliseconds, suddenly it becomes a feature. Even if a competitor technically has the feature, you make it practical.

All this is just another way of saying: Be specific (p. 1439), avoid buzzwords (p. 627), be fully committed to your ideal customer (p. 317) and what they value (p. 259), and tell the truth.

It's a critical component of having a great strategy (p. 489), and great positioning.

Chapter 156:
Idiot! Buying SmartBear.com

Being slow cost me $2000. **IDIOT!**

Yesterday I bought `smartbear.com`.

Smart Bear started as a sole proprietorship.[1464] I switched it to a single-member LLC[1465] in Feb 2003, mostly just to see what it was like. (Single-member LLC's and sole proprietorships have identical tax consequences—none.)

At the time Code Historian[1466] and Code Reviewer[1467] were shareware and the company name didn't seem to matter. After all, I had `http://codehistorian.com`, and that's better! So I didn't register `smartbear.com`.

Let me repeat that. I didn't register `smartbear.com`.

I could tell you I didn't have any money to spend on such things, which was true, but that's not an excuse, it's cheap. I could tell you I didn't think things would go so well, which was true, but that's not an

excuse, it would have been cheap insurance. I could tell you I didn't think anyone was paying attention to little ol' me, which was true, but that's not an excuse, the domain trollers watch the chamber of commerce records and grab anything they can find.

There's no excuse.

By the time I came to my senses, a squatter had taken the domain. For a while the page was just broken. Then the page simply said the domain was for sale. $2000. I ended up getting `smartbearsoftware.com`.

The way you pronounce "`smartbearsoftware.com`" over the phone is:

> "Smart Bear Software. One word. That's "bear" as in the animal. No, Smart Bear Software. You have to have the software. (pause) Yeah, I know, that's taken by a squatter. (pause) two thousand dollars. (pause) Yeah, we really should. It's the principle of the thing though."

And it really was. I offered the guy $1000. Rejected. I sent a lawyer-letter. Ignored. I asked the lawyer how much it would cost to actually pursue this. Minimum $2500. (Oh. That's how the price is set.)

I pointed out the trademark. He responded by saying it wasn't a registered trademark. I started the process of registering all our trademarks, but years later we're in the last stage of the process but still waiting.

I was goaded into this by the other Bears. "Just get the domain. We know it sucks. We know it's unfair. But stop making us type 'software' all the time like chumps."

The final straw was a conversation with Chris Boyd, co-founder of the Best Little ISP on the Planet, Midas Networks.[1468] I told him about the domain. "Yeah," he said, "you just have to pay those guys." Chris always tells it straight. That's when I know I was sunk.

So I did it.

References

1. "Complicated Cartoon #7670", *Andertoons* — https://andertoons.com/complicated/cartoon/7670/easy-follow-rules-more-difficult-implement-than-youd-think
2. "Reboot: Leadership and the Art of Growing Up", *Reboot* — https://www.reboot.io/reboot-leadership-and-the-art-of-growing-up/
3. "Marriage Cartoon #8849", *Andertoons* — https://andertoons.com/marriage/cartoon/8849/robe-sorcery-potions-use-coaster
4. https://dilbertblog.typepad.com/the_dilbert_blog/2007/07/career-advice.html
5. "How to Get Startup Ideas" — https://paulgraham.com/startupideas.html
6. https://twitter.com/randfish/status/1647666512290013184?s=20
7. https://www.linkedin.com/feed/update/urn:li:activity:7053396708713459712/?commentUrn=urn%3Ali%3Acomment%3A%28activity%3A7053396708713459712%2C7053504417685274624%29&dashCommentUrn=urn%3Ali%3Afsd_comment%3A%287053504417685274624%2Curn%3Ali%3Aactivity%3A7053396708713459712%29
8. https://twitter.com/nathanjpowellux/status/1656536138901778432?s=12&t=mnlAhEmLkMKMeX_fkP_pzw
9. "I try to stay away from hyperbole, but I'm struggling to when it comes to this post ..." by Brian Rhea, *LinkedIn* — https://www.linkedin.com/feed/update/urn:li:activity:7053766029830754304/
10. "I really enjoyed this write-up by Jason Cohen." by Aaron Roy, *LinkedIn* — https://www.linkedin.com/posts/aaronmichaelroy_excuse-me-is-there-a-problem-activity-7096617697798434816-1U22/?utm_source=share&utm_medium=member_desktop
11. *Twitter* — https://x.com/TheWenJie/status/1735476053135737109?s=20
12. https://twitter.com/advany/status/1745429907788689516
13. https://twitter.com/advany/status/1746873616354926881
14. "Hope Cartoon #7605", *Andertoons* — https://andertoons.com/hope/cartoon/7605/well-its-not-the-worst-ive-seen
15. "Overcoming the Pervasive Analytical Blunder of Strategists" by Roger Martin, *Medium* — https://rogermartin.medium.com/overcoming-the-pervasive-analytical-blunder-of-strategists-a5a4c52289b4
16. "Interview Cartoon #7030", *Andertoons* — https://andertoons.com/interview/cartoon/7030/so-tell-me-a-little-bit-of-what-you-think-i-want-to-hear-about-yourself
17. "Bob Moesta on unpacking customer motivations with Jobs-to-be-Done" by Des Traynor, *The Intercom Blog* — https://www.intercom.com/blog/podcasts/bob-moesta-on-unpacking-customer-motivations-with-jobs-to-be-done/

18. *Twitter* – https://x.com/JustinQuda/status/1734646310978879821?s=20
19. "Pick one and own it" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/one-benefit/
20. *Twitter* – https://x.com/OrionSeven/status/1732459281549570522?s=20
21. "Fish Cartoon #5228", *Andertoons* – https://andertoons.com/fish/cartoon/5228/im-looking-for-something-in-small-pond
22. https://twitter.com/flybayer/status/1725298305591132472
23. *Twitter* – https://x.com/Timb03/status/1762691202028028089?s=20
24. *Twitter* – https://x.com/nafetswirth/status/1847580235023175815?s=12&t=mnlAhEmLkMKMeX_fkP_pzw
25. "Sex Cartoon #1906", *Andertoons* – https://andertoons.com/sex/cartoon/1906/no-youre-thinking-of-cupid-im-one-night-stand-fairy
26. ""ROI" is the wrong way to sell your product" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/roi-selling/
27. "Worth Cartoon #7782", *Andertoons* – https://andertoons.com/worth/cartoon/7782/youll-never-be-part-of-a-royal-flush-me-neither-thats-ok
28. "What a startup does to you. Or: A celebration of new life" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/startup-life/
29. "WP Engine is rated "Excellent" with 4.4 / 5 on" by Admin, *Trustpilot* – https://www.trustpilot.com/review/wpengine.com
30. "Simon Høiberg", *logo* – https://www.simonhoiberg.com/
31. "FeedHive", *Create content at scale* – https://www.feedhive.com/
32. "Threads" – https://www.threads.com/@simonhoiberg/post/C04BZ4ktV3F
33. "Business Startups Advice & Fundraising Advice", *Shockwave Innovations* – https://shockwaveinnovations.com/
34. *Rowan Udell* – https://blog.rowanudell.com/
35. https://www.linkedin.com/in/sathyanands/
36. "Five Artists Who Refused to Sell Out •" by James Tait, *Howl & Echoes* – https://howlandechoes.com/2014/11/five-artists-refused-sell/
37. "SmartBear Software - The confidence behind your code.", *SmartBear Software* – https://smartbear.com/
38. ""Rich versus King": The Core Concept" by Noam Wasserman – https://www.noamwasserman.com/rich-versus-king-the-core-concept/
39. "Chiles Rellenos de Carne Molida" by Carissa, *Thrift and Spice* – https://thriftandspice.com/chiles-rellenos-de-carne-molida/
40. "The Guerrilla Guide to Interviewing (version 3.0)" by Joel Spolsky, *Joel on Software* – https://www.joelonsoftware.com/2006/10/25/the-guerrilla-guide-to-interviewing-version-30/
41. "A Tradeshow Checklist, born of experience" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/tradeshow-tips-checklist/
42. "Creating 'Naked Businesses'" by Mark T Littlewood, *Business of Software* – https://businessofsoftware.org/2012/07/jason-cohen-asmartbear-founder-of-smart-bear-software-talks-on-creating-naked-businesses/
43. *Twitter* – https://x.com/RealJoshLong/status/1815978972335034661
44. https://twitter.com/Codie_Sanchez/status/1739729553143058700
45. "What a startup does to you. Or: A celebration of new life" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/startup-life/
46. "Devil Cartoon #6064", *Andertoons* – https://andertoons.com/devil/cartoon/6064/boy-what-day-huh
47. "Meeting Cartoon #512", *Andertoons* – https://andertoons.com/meeting/cartoon/512/so-as-you-can-see-customer-satisfaction-up-considerably-since-phasing-out-complaint-forms

48. *Amazon* – https://www.amazon.com/Effortless-Experience-Conquering-Battleground-Customer/dp/1591845815
49. "Subscription Cartoon #6535", *Andertoons* – https://andertoons.com/subscription/cartoon/6535/instead-of-allowance-how-about-subscription-model
50. "Business Cartoon #7278", *Andertoons* – https://andertoons.com/business/cartoon/7278/restructuring-im-tom-tom-anne-anne-copier-copier-in-charge
51. https://twitter.com/andrewwatson/status/1641885868754456591
52. "What's a Good Clickthrough Rate? New Benchmark Data for Google AdWords" by Kristina Volovich, *HubSpot* – https://blog.hubspot.com/agency/google-adwords-benchmark-data
53. "What's a good conversion rate in B2B SaaS and beyond?" by Enricko Lukman, *ContentGrip* – https://www.contentgrip.com/conversion-rate-business-benchmark/
54. "What is a Good Conversion Rate and How to Improve It" by Adoric Team, *Adoric Blog* – https://adoric.com/blog/what-is-a-good-conversion-rate-2020/
55. https://twitter.com/CoryMckane/status/1639664575686619138?s=20
56. https://twitter.com/r00k/status/1639667146409009153?s=20
57. https://twitter.com/timb03/status/1637211704093478912?s=12&t=mnlAhEmLkMKMeX_fkP_pzw
58. https://twitter.com/shl/status/1639669362885021696?s=20
59. https://twitter.com/jdnoc/status/1639669338587164674?s=20
60. "I have a theory that it _typically_ takes 2 years to get to 1,000 _paying_ customers (at ≥$30/mo), even for companies who "hyper-scale."" by Jason Cohen, *LinkedIn* – https://www.linkedin.com/feed/update/urn:li:activity:7045425924321767425?commentUrn=urn%3Ali%3Acomment%3A%28activity%3A7045425924321767425%2C7045431150504280064%29&replyUrn=urn%3Ali%3Acomment%3A%28activity%3A7045425924321767425%2C7045432435030822912%29&dashCommentUrn=urn%3Ali%3Afsd_comment%3A%287045431150504280064%2Curn%3Ali%3Aactivity%3A7045425924321767425%29&dashReplyUrn=urn%3Ali%3Afsd_comment%3A%287045432435030822912%2Curn%3Ali%3Aactivity%3A7045425924321767425%29
61. https://twitter.com/hmps_/status/1639687300777951233?s=20
62. https://twitter.com/jfriedlaender/status/1639766616270602240?s=46&t=mnlAhEmLkMKMeX_fkP_pzw
63. https://twitter.com/danielkempe/status/1639894339521327104?s=20
64. https://twitter.com/AlecEllin/status/1640008710507360257?s=20
65. https://twitter.com/aprilzero/status/1639810561624899584?s=46&t=mnlAhEmLkMKMeX_fkP_pzw
66. https://twitter.com/thedannorris/status/1639842020570308609?s=46&t=mnlAhEmLkMKMeX_fkP_pzw
67. "How Do You Get Your First 1,000 Customers?" by brett fox, *The Startup* – https://medium.com/swlh/how-do-you-get-your-first-1-000-customers-dd3b3968212a
68. "About Ghost - The Open Source Publishing Platform", *Ghost - The Professional Publishing Platform* – https://ghost.org/about/
69. https://twitter.com/AngeloRobertsJr/status/1640103585550614529?s=20
70. https://twitter.com/masonarnold/status/1639965501118922752?s=20
71. https://twitter.com/alxberman/status/1639900347136380928?s=20
72. https://twitter.com/cemper/status/1639728089440681984?s=46&t=mnlAhEmLkMKMeX_fkP_pzw
73. "Schumpeter on Strategy" by Kirk Willard, *Reaction Wheel* – https://reactionwheel.net/2019/01/schumpeter-on-strategy.html
74. "Safety Cartoon #6712", *Andertoons* – https://andertoons.com/safety/cartoon/6712/sign-reads-162-days-without-addressing-the-real-problem

75. "OpenAI CEO Sam Altman on GPT-4 & the A.I. Arms Race", *Apple Podcasts* – https://podcasts.apple.com/us/podcast/openai-ceo-sam-altman-on-gpt-4-the-a-i-arms-race/id1643307527?i=1000605522804
76. "The Global PPC Click Fraud Report 2020-21" by PPC Protect, *Search Engine Journal* – https://www.searchenginejournal.com/the-global-ppc-click-fraud-report-2020-21/391493/
77. "How Companies Are Hacked via Malicious Javascript Code?" by Ebubekir Büber, *ITNEXT* – https://itnext.io/how-companies-are-hacked-via-malicious-javascript-code-12aa82560bdc?gi=a15fa6e7ec22
78. "Webvan", *Wikipedia* – https://en.wikipedia.org/wiki/Webvan
79. "Christmas Cartoon #7423", *Andertoons* – https://andertoons.com/Christmas/cartoon/7423/naughty-children-who-dont-get-any-toys-an-underserved-niche
80. "Home", *Capital Factory* – https://capitalfactory.com/
81. "Bear", *Bear Markdown Notes* – https://bear.app/
82. "Remember The Milk: Online to-do list and task management" – https://www.rememberthemilk.com/
83. "Reddit - The heart of the internet" by Tom-Solid, *Log In* – https://www.reddit.com/r/Evernote/comments/11ir46v/evernote_is_increasing_their_pricing_by_nearly_50/
84. "Thanksgiving Cartoon #6158", *Andertoons* – https://andertoons.com/thanksgiving/cartoon/6158/we-found-problem-all-of-your-internal-organs-are-in-this-little-bag
85. "Advertising Cartoon #106", *Andertoons* – https://andertoons.com/advertising/cartoon/106/its-not-that-i-dont-like-you-ted-you-just-dont-fit-my-target-demographic
86. "SaaS Plateau Example", *Summit* – https://usesummit.com/free-calculators/ade496/saas-plateau-forecaster/?conversion_rate=25&arpc=100&churn_rate=7&mrr=15k&new_trials=80&_start=2023-03-01&_end=2025-12-30&_resolution=month&_title=SaaS+Plateau+Example&_description=Based+on+the+example+in+https%3A%2F%2Flongform.asmartbear.com%2Fproblem%2F
87. "Summit" by Ruben GamezFounder, *Summit logo* – https://usesummit.com/
88. "My Top 10 Mistakes In 10 Years: Gainsight CEO Nick Mehta" by Jason M. Lemkin 🦄, *The Secrets To Scaling in The Age of AI* – https://cloud.substack.com/p/my-top-10-mistakes-in-10-years-gainsight?open=false
89. "Shopify has a growing problem with customer retention, Globe data study shows" by Chris Hannay, *The Globe and Mail* – https://www.theglobeandmail.com/business/article-shopify-customer-retention-problem/
90. "The Best React-Based Framework", *Gatsby* – https://www.gatsbyjs.com/
91. https://twitter.com/calcsam/status/1645045534326145025
92. https://www.businessdit.com/how-many-businesses-are-there-in-the-world/
93. "20 Incredible Online Business Statistics [2023]" by Jack Flynn, *Zippia* – https://www.zippia.com/advice/online-business-statistics/
94. "Usage Statistics and Market Share of Content Management Systems, June 2025" – https://w3techs.com/technologies/overview/content_management
95. "Kit: Email-First Operating System for Creators (formerly ConvertKit)", *Kit* – https://kit.com/
96. "Internet and social media users in the world 2025" by Ani Petrosyan, *Statista* – https://www.statista.com/statistics/617136/digital-population-worldwide/?__sso_cookie_checker=failed
97. "Problem Cartoon #8144", *Andertoons* – https://andertoons.com/problem/cartoon/8144/why-no-one-likes-math-branding-issue-everythings-a-problem
98. "Five years later: Five ways to build a $100 million SaaS business" by Christoph Janz – https://christophjanz.blogspot.com/2019/04/five-years-later-five-ways-to-build-100.html
99. "The mid-market briar patch" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/mid-market/
100. https://twitter.com/DanielZarick

101. https://twitter.com/KimStacks
102. https://twitter.com/mattwensing
103. https://twitter.com/calcsam
104. https://twitter.com/Tony_Meijer
105. https://twitter.com/wfjackson3
106. "Startup Viability Score (Interactive Calculator)" – https://www.codelantis.com/blog/startup-viability-score
107. "Complicated Cartoon #7670", *Andertoons* – https://andertoons.com/complicated/cartoon/7670/easy-follow-rules-more-difficult-implement-than-youd-think
108. "Minimum viable product", *Wikipedia* – https://en.wikipedia.org/wiki/Minimum_viable_product
109. https://twitter.com/reidhoffman
110. "Darwinian advice for "Going Viral"" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/going-viral/
111. "Buffer Blog - Thoughts on Social Media & Online Marketing" by Kirsti Lang, *Buffer: All-you-need social media toolkit for small businesses* – https://buffer.com/resources/
112. "Blog", *Social Media Marketing Tips, Trends & Best Practices at EdgarInk* – https://meetedgar.com/blog
113. "HubSpot Blog" by Maxwell Iskiev, *HubSpot* – https://blog.hubspot.com/
114. "File:Maslow's Hierarchy of Needs2.svg - Wikimedia Commons", *Wikipedia* – https://commons.wikimedia.org/wiki/File:Maslow%27s_Hierarchy_of_Needs2.svg
115. "Maslow's hierarchy of needs", *Wikipedia* – https://en.wikipedia.org/wiki/Maslow%27s_hierarchy_of_needs
116. "Threads" – https://www.threads.com/@lucid_voyager/post/DAMOUqCyB3J
117. "Threads" – https://www.threads.com/@kathyqian/post/DAMSqhTS6KD
118. "Threads" – https://www.threads.com/@khurrum.carpediem/post/DAMRiiFOTjl
119. "BNN Bloomberg - Canada Business News, TSX Today, Oil and Energy Prices", *BNN* – https://www.bnnbloomberg.ca/
120. "How Slack Became The Fastest-Growing Enterprise Software Ever" by John Koetsier, *Forbes* – https://www.forbes.com/sites/johnkoetsier/2018/11/30/how-slack-became-the-fastest-growing-enterprise-software-ever/
121. "Not all Daily Active Users are created equal: Work is fueled by true engagement", *Slack* – https://slack.com/blog/news/work-is-fueled-by-true-engagement
122. "Slack vs. Teams" by Catalin Alaci, *BEMO* – https://www.bemopro.com/cybersecurity-blog/slack-vs.-teams
123. https://www.investx.com/deals/dropbox/growth-strategy
124. https://saasscout.com/statistics/dropbox-statistics/
125. "Why Trello Failed to Build a $1 Billion+ Business" by Hiten Shah, *Medium* – https://medium.com/@hnshah/why-trello-failed-to-build-a-1-billion-business-e1579511d5dc
126. "S-1" – https://www.sec.gov/Archives/edgar/data/1759509/000119312519059849/d633517ds1.htm
127. "Linguistic relativity", *Wikipedia* – https://en.m.wikipedia.org/wiki/Linguistic_relativity
128. https://www.hubspot.com/hubfs/Investor%20Presentation%20Q121%20%281%29.pdf
129. "The iPod Is Dead, And These Photos Will Make You Miss Yours" by Drew Guarini, *HuffPost* – https://www.huffpost.com/entry/ipod-sales_n_4680000
130. "Infographic: Apple's Constant Search for the Next Big Thing" by Felix Richter, *Statista Daily Data* – https://www.statista.com/chart/17862/apples-annual-revenue-by-operating-segment/?__sso_cookie_checker=failed
131. "Predictable Growth Decay in SaaS Companies" by Andy Vitus, *Scale Venture Partners* – https://www.scalevp.com/insights/predictable-growth-decay-in-saas-companies/

132. "Wordle creator overwhelmed by global success of hit puzzle" by Rachel Hall, *The Guardian* – https://www.theguardian.com/games/2022/jan/11/wordle-creator-overwhelmed-by-global-success-of-hit-puzzle

133. "The New York Times Buys Wordle (Published 2022)" by Marc Tracy, *The New York Times* – https://www.nytimes.com/2022/01/31/business/media/new-york-times-wordle.html

134. "Twitter MAU in the United States 2019" by Stacy Jo Dixon, *Statista* – https://www.statista.com/statistics/274564/monthly-active-twitter-users-in-the-united-states/?__sso_cookie_checker=failed

135. "eBay Revenue and Usage Statistics (2025)" by David Curry, *Business of Apps* – https://www.businessofapps.com/data/ebay-statistics/

136. "Smartphone sales by OS worldwide 2009-2018" by Petroc Taylor, *Statista* – https://www.statista.com/statistics/266219/global-smartphone-sales-since-1st-quarter-2009-by-operating-system/?__sso_cookie_checker=failed

137. "Desktop vs Mobile Market Share Worldwide", *StatCounter Global Stats* – https://gs.statcounter.com/platform-market-share/desktop-mobile/worldwide/

138. https://www.businessinsider.com/how-facebook-plans-to-improve-messenger-in-2018-1

139. "Facebook lost daily active users for the first time ever in the U.S. and Canada" by Kurt Wagner, *Vox* – https://www.vox.com/2018/1/31/16957122/facebook-daily-active-user-decline-us-canda-q4-earnings-2018

140. "Facebook just published its 2019 performance report, The social network boasts 1.66 Billion Daily Active Users" by agha ali, *Digital Information World* – https://www.digitalinformationworld.com/2020/01/facebook-q4-2019-1657-million-dau-2498-million-mau.html

141. "Internet" by Max Roser, *Our World in Data* – https://ourworldindata.org/internet

142. "Netflix's original and international content are fueling its growth" by Mariel Soto Reyes, *Business Insider* – https://www.businessinsider.com/netflixs-original-international-content-fuel-q1-growth-2019-4

143. "Daily Data" by Felix Richter, *Statista Daily Data* – https://www.statista.com/chartoftheday/?__sso_cookie_checker=failed

144. https://twitter.com/stats_feed/status/1609104967935164417?s=12&t=XEDWR_LNgM1uzUNNrX6WaA

145. "Loon - A Google X Moonshot" by X, the moonshot factory, *X* – https://x.company/projects/loon/

146. https://www.facebook.com/connectivity/solutions/free-basics

147. "'It's digital colonialism': how Facebook's free internet service has failed its users" by Olivia Solon, *The Guardian* – https://www.theguardian.com/technology/2017/jul/27/facebook-free-basics-developing-markets

148. "Global Meta advertising revenue 2024" by Stacy Jo Dixon, *Statista* – https://www.statista.com/statistics/271258/facebooks-advertising-revenue-worldwide/?__sso_cookie_checker=failed

149. "Speed Of Implementation and the Law Of Shitty Clickthroughs" by Taylor, *Taylor Pearson* – https://taylorpearson.me/ctr/

150. "Which Half of my Advertising is Wasted – and It Is Only Half?" by Darren Woolley, *MediaVillage* – https://www.mediavillage.com/article/which-half-of-my-advertising-is-wasted-and-it-is-only-half/

151. "News Cartoon #6605", *Andertoons* – https://andertoons.com/news/cartoon/6605/son-comes-time-in-life-when-need-to-decide-news-to-believe

152. "What a startup does to you. Or: A celebration of new life" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/startup-life/

153. "Easy to criticize, hard to create" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/criticize-create/

154. "VC Fund Returns Are More Skewed Than You Think", *VC Adventure* – https://sethlevine.com/archives/2020/10/vc-fund-returns-are-more-skewed-than-you-think.html

155. https://twitter.com/anyuser/status/1742484259213029599

156. https://twitter.com/anyuser/status/1742518005836271628

157. "Daanlo (@Daanlo@mastodon.social)", *Mastodon* – https://mastodon.social/@Daanlo/111727865682080974

158. "When you're in the thick of things, it isn't working yet but it's not totally a failure, how do you know whether" by Jason Cohen, *LinkedIn* – https://www.linkedin.com/feed/update/urn:li:activity:7148238290591125505?commentUrn=urn%3Ali%3Acomment%3A%28activity%3A7148238290591125505%2C7148314532505972736%29&dashCommentUrn=urn%3Ali%3Afsd_comment%3A%287148314532505972736%2Curn%3Ali%3Aactivity%3A7148238290591125505%29

159. https://twitter.com/anyuser/status/1742477070662328629

160. *Twitter* – https://x.com/sachinrekhi/status/1742594429884694858?s=20

161. https://twitter.com/anyuser/status/1742501512688070677

162. *Twitter* – https://x.com/mibenz95/status/1742660395960025508?s=20

163. *Twitter* – https://x.com/nick_res_real/status/1742629476079444371?s=20

164. https://twitter.com/nurijanian/status/1742693171833000107

165. "When You Should Quit" by Alex Finn, *1% Better* – https://www.1percentbetter.io/p/quit-a2b1

166. https://twitter.com/Liscoomi/status/1742719163766886544

167. https://twitter.com/anyuser/status/1742523382158016600

168. "Andrew Woods (@awoodsnet@phpc.social)", *PHP Community on Mastodon* – https://phpc.social/@awoodsnet/111695017297210504

169. https://twitter.com/anyuser/status/1742536876156748116

170. "Car Cartoon #3220", *Andertoons* – https://andertoons.com/car/cartoon/3220/i-like-it-but-im-looking-for-more-of-status-symbol-any-way-you-can-double-price

171. ""ROI" is the wrong way to sell your product" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/roi-selling/

172. "Baseball Cartoon #6857", *Andertoons* – https://andertoons.com/baseball/cartoon/6857/traded-for-some-big-data-two-spreadsheets-and-an-algorithm

173. "Enrico Fermi" by Argonne National Laboratory, *Flickr* – https://www.flickr.com/photos/35734278@N05/5039459604

174. "Argonne National Laboratory" by Argonne National Laboratory, *Flickr* – https://www.flickr.com/photos/35734278@N05

175. "Deed - Attribution-NonCommercial-ShareAlike 2.0 Generic", *Creative Commons* – https://creativecommons.org/licenses/by-nc-sa/2.0/?ref=openverse

176. "Trinity Test, July 16, 1945, Eyewitness Accounts" by Gene Dannen, *Enrico Fermi* – https://www.dannen.com/decision/fermi.html

177. "Gas Cartoon #5206", *Andertoons* – https://andertoons.com/gas/cartoon/5206/gas-pumps-in-front-of-prophecy-gas-read-fulfilling-and-self-fulfilling

178. "The Guerrilla Guide to Interviewing" by Joel Spolsky, *Joel on Software* – https://www.cse.unr.edu/~sushil/researchAdvice/GIV1.html

179. "Los Angeles County, California", *Wikipedia* – https://en.wikipedia.org/wiki/Los_Angeles_County,_California

180. "Density, Car Ownership, and What It Means for the Future of Los Angeles - Streetsblog Los Angeles" by Damien Newton, *Streetsblog Los Angeles home* – https://la.streetsblog.org/2010/12/13/density-car-ownership-and-what-it-means-for-the-future-of-los-angeles

181. "How many gas stations in los angeles?" by Harry Chen, *Traveler's blog* – https://lastfiascorun.com/brazil/how-many-gas-stations-in-los-angeles.html

182. https://en.wikipedia.org/wiki/Fibonacci_scale_(agile

183. https://wpengine.com/atlas

184. "Meeting Cartoon #6039", *Andertoons* – https://andertoons.com/meeting/cartoon/6039/meeting-person-realizes-hes-only-brought-his-c-game

185. "Insect Cartoon #4408", *Andertoons* — https://andertoons.com/insect/cartoon/4408/so-where-do-you-see-yourself-in-ten-minutes

186. "Sales Cartoon #7195", *Andertoons* — https://andertoons.com/sales/cartoon/7195/recommended-against-upping-game-while-taking-to-next-level

187. "News Cartoon #4033", *Andertoons* — https://andertoons.com/news/cartoon/4033/reaction-to-news-was-mixed-largely-because-we-asked-more-than-one-person

188. "The best Vanguard bond funds for UK investors - Occam Investing" by Occam, *Occam Investing* - — https://occaminvesting.co.uk/the-best-vanguard-bond-funds-for-uk-investors/

189. "Butterfly effect", *Wikipedia* — https://en.wikipedia.org/wiki/Butterfly_effect

190. "Butterfly Cartoon #8129", *Andertoons* — https://andertoons.com/butterfly/cartoon/8129/flap-wings-crazy-things-caused-sounds-lot-responsibility

191. https://research-doc.credit-suisse.com/docView?language=ENG&format=PDF&source_id=csplusresearchcp&document_id=1052373521&serialid=6mlgSOPrsTMVqNGLKrbx-Ph5WhdLmUNmf90sj0LLxV%2F8%3D&cspId=null

192. "Infographic: How Successful Is Tesla?" by Florian Zandt, *Statista Daily Data* — https://www.statista.com/chart/26705/yearly-net-income-and-revenue-of-tesla/?__sso_cookie_checker=failed

193. "Reddit - The heart of the internet" by CalebWetherell, *Log In* — https://www.reddit.com/r/chess/comments/10no1pk/tata_steel_masters_predictions_after_round_12_the/

194. "Reddit - The heart of the internet" by TheKitof, *Log In* — https://www.reddit.com/r/dataisbeautiful/comments/zjeouy/oc_2022_world_cup_probabilities_of_final_victory/

195. https://commons.wikimedia.org/wiki/File:Game_Neverending_2008.png

196. "Butterfly milking and pig nibbling: building the strange world of Glitch" by Jeremy Reimer, *Ars Technica* — https://arstechnica.com/gaming/2011/11/glitch-the-battle-to-build-a-massive-multiplayer-game-without-combat/

197. https://www.glitchthegame.com/closing/

198. "We Don't Sell Saddles Here" by Stewart Butterfield, *Medium* — https://medium.com/@stewart/we-dont-sell-saddles-here-4c59524d650d

199. "Content Marketing Case Study: How Intercom Built a $50MM ARR Empire" by Tyler Hakes, *Optimist* — https://www.yesoptimist.com/intercom-growth-strategy-teardown/

200. "VISUAL EDIT: WhatsApp is killing SMS" by Mail Today Comment, *Daily Mail* — https://www.dailymail.co.uk/indiahome/indianews/article-3037554/VISUAL-EDIT-WhatsApp-killing-SMS.html

201. "Antifragility", *Wikipedia* — https://en.wikipedia.org/wiki/Antifragility

202. *Amazon* — https://www.amazon.com/Antifragile-Things-That-Disorder-Incerto/dp/0812979680/

203. "The Five Principles of Effectuation", *Effectuation Extended Logo (200 × 82 px)* — https://effectuation.org/the-five-principles-of-effectuation

204. "Why Emotionally Intelligent People Embrace the 2-Way-Door Rule to Make Better and Faster Decisions" by Jeff Haden, *Inc* — https://www.inc.com/jeff-haden/why-emotionally-intelligent-people-embrace-2-way-doors-rule-to-make-better-faster-decisions.html

205. "OODA loop", *Wikipedia* — https://en.wikipedia.org/wiki/OODA_loop

206. "Welcome", *Product Talk* — https://www.producttalk.org/

207. "Continuous delivery", *Wikipedia* — https://en.wikipedia.org/wiki/Continuous_delivery

208. "Insurance Cartoon #7723", *Andertoons* — https://andertoons.com/insurance/cartoon/7723/huff-puff-we-move-with-brother-all-split-insurance-money

209. "Apple's Dueling iPhone OS Projects", *Daring Fireball* — https://daringfireball.net/linked/2022/01/05/apples-dueling-iphone-os-projects

210. "Unpacking growth with Andy Johns, Partner at Unusual Ventures" by Erasmus Elsner, CFA, *Sand Hill Road* — https://medium.com/sand-hill-road/unpacking-growth-with-andy-johns-partner-at-unusual-ventures-79778fc7bdd8

211. "A generic foresight process framework" by Authors, *Emerald Insight* — https://www.emerald.com/insight/content/doi/10.1108/14636680310698379/full/html

212. "Theory of Change and the Futures Cone." by Sjef, *Sjef van Gaalen* – https://sjef.nu/theory-of-change-and-the-futures-cone/
213. https://au.linkedin.com/in/adamdangerbrock
214. https://twitter.com/amyhoy
215. https://twitter.com/danielzarick
216. https://twitter.com/bigdeechi
217. https://twitter.com/JJJ
218. "Entrepreneurial Engineer" by KimSia Sim, *Substack* – https://www.entrepreneurial.engineer/
219. https://twitter.com/mothcamp
220. https://twitter.com/phuggins
221. https://twitter.com/prasanna_says
222. https://twitter.com/RhysJeffery2
223. https://twitter.com/sethchasin
224. https://twitter.com/Tony_Meijer
225. "Videos: The 8th Habit", *FranklinCovey* – https://www.franklincovey.com/resources/videos/the-8th-habit/
226. https://www.integrativenutrition.com/big-rocks
227. "Work Cartoon #8717", *Andertoons* – https://andertoons.com/work/cartoon/8717/sisyphus-treadmill-working-from-home
228. https://twitter.com/housecor/status/1552324251876593664?s=21
229. "A quote from Good to Great" by Good to Great: Why Some Companies Make the Leap... and Others Don't, *Share on Twitter* – https://www.goodreads.com/quotes/701885-good-is-the-enemy-of-great-and-that-is-one
230. "Sunk Costs: An invisible, pervasive peril" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/sunk-costs/
231. "Jeff Bezos: There are 2 types of decisions to make, and don't confuse them" by Matt Rosoff, *Business Insider* – https://www.businessinsider.com/jeff-bezos-on-type-1-and-type-2-decisions-2016-4
232. "Police Cartoon #8798", *Andertoons* – https://andertoons.com/police/cartoon/8798/scissors-rock-lineup
233. "Chicken Cartoon #8651", *Andertoons* – https://andertoons.com/chicken/cartoon/8651/start-with-some-small-pebbles
234. "Eisenhower Matrix", *ProductPlan* – https://www.productplan.com/glossary/eisenhower-matrix/
235. "Pirate Cartoon #8822", *Andertoons* – https://andertoons.com/pirate/cartoon/8822/plundering-small-part-of-job
236. "Customer development", *Wikipedia* – https://en.wikipedia.org/wiki/Customer_development
237. "Know Your Customers' "Jobs to Be Done"" by Clayton M. Christensen, *Harvard Business Review* – https://hbr.org/2016/09/know-your-customers-jobs-to-be-done
238. "Retroactive continuity", *Wikipedia* – https://en.wikipedia.org/wiki/Retroactive_continuity
239. "Confirmation bias", *Wikipedia* – https://en.wikipedia.org/wiki/Confirmation_bias
240. "Talk Cartoon #7885", *Andertoons* – https://andertoons.com/talk/cartoon/7885/thing-youre-here-to-talk-about-talk-about-that-a-little
241. "Vetting a startup (or two): The systematic birth of @WPEngine" by Jason, *@ASmartBear* – https://blog.asmartbear.com/vetting-startup-ideas/
242. "40 Tips for B2B Customer Development Interviews" by Sean Murphy, *SKMurphy, Inc.* – https://www.skmurphy.com/blog/2020/01/30/40-tips-for-b2b-customer-development-interviews/

243. *Twitter* — https://x.com/skmurphy

244. "12 Tips for Early Customer Development Interviews (Revision 3)", *giffconstable.com* — https://giffconstable.com/2012/12/12-tips-for-early-customer-development-interviews-revision-3/

245. "11 Customer Development Anti-Patterns", *giffconstable.com* — https://giffconstable.com/2013/06/11-customer-development-anti-patterns/

246. https://twitter.com/giffco

247. "Customer Interviews: How to Recruit, What to Ask, and How to Synthesize What You Learn" by Teresa Torres, *Product Talk* — https://www.producttalk.org/2022/12/customer-interviews/

248. https://twitter.com/ttorres

249. *Amazon* — https://www.amazon.com/Continuous-Discovery-Habits-Discover-Products/dp/1736633309

250. https://www.momtestbook.com/

251. "Ignore Cartoon #7869", *Andertoons* — https://andertoons.com/ignore/cartoon/7869/youve-given-me-a-lot-to-forget-about-after-you-leave

252. "Bird Cartoon #7353", *Andertoons* — https://andertoons.com/bird/cartoon/7353/knew-it-was-bad-stopped-wanting-cracker-started-needing-one

253. "Historical quarterly trends in the usage statistics of content management systems, June 2025" — https://w3techs.com/technologies/history_overview/content_management/all/q

254. "Clay Christensen's Milkshake Marketing" by Clayton Christensen, *Harvard Business School* — https://www.library.hbs.edu/working-knowledge/clay-christensens-milkshake-marketing

255. "Five whys", *Wikipedia* — https://en.wikipedia.org/wiki/Five_whys

256. "WP Engine, the Smarter Way to Host WordPress", *WP Engine* — https://wpengine.com/

257. "What a startup does to you. Or: A celebration of new life" by Jason Cohen, *A Smart Bear* — https://longform.asmartbear.com/startup-life/

258. "Official Website of Tony Robbins: Personal & Business Results Coach", *Tony Robbins* — https://www.tonyrobbins.com/

259. "Find and Hire Top Speakers for Your Events" by sm_content_admin, *SpeakerMatch* — https://speakermatch.com/

260. "Marketing Cartoon #7214", *Andertoons* — https://andertoons.com/marketing/cartoon/7214/this-is-tom-he-creates-awareness

261. "Market Cartoon #7965", *Andertoons* — https://andertoons.com/market/cartoon/7965/sounds-like-a-lot-of-work-could-we-make-a-big-mess-instead

262. *Amazon* — https://www.amazon.com/Blue-Ocean-Strategy-Uncontested-Competition/dp/1591396190/ref=asc_df_1591396190/?tag=hyprod-20&linkCode=df0&hvadid=312175933381&hvpos=&hvnetw=g&hvrand=11171918790451270235&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9028297&hvtargid=pla-522339108124&psc=1&tag=&ref=&adgrpid=60258871817&hvpone=&hvptwo=&hvadid=312175933381&hvpos=&hvnetw=g&hvrand=11171918790451270235&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9028297&hvtargid=pla-522339108124

263. "Shopify has a growing problem with customer retention, Globe data study shows" by Chris Hannay, *The Globe and Mail* — https://www.theglobeandmail.com/business/article-shopify-customer-retention-problem/

264. "Shopify Announces Fourth-Quarter and Full-Year 2022 Financial Results", *Shopify* — https://www.shopify.com/news/shopify-announces-fourth-quarter-and-full-year-2022-financial-results

265. "Sales Cartoon #7444", *Andertoons* — https://andertoons.com/sales/cartoon/7444/ok-lets-not-get-into-profit-shaming-here

266. "CEO: 5,000-percent drug price hike "not excessive at all"", *CBS News* — https://www.cbsnews.com/news/turing-pharmaceuticals-ceo-martin-shkreli-defends-5000-percent-price-hike-on-daraprim-drug/

267. "Smirking pharma-villain Martin Shkreli pleads the fifth, calls US lawmakers 'imbeciles'" by Reuters, *South China Morning Post* – https://www.scmp.com/news/world/article/1909685/smirking-pharma-villain-shkreli-pleads-fifth-calls-lawmakers-imbeciles

268. "How 4 drug companies rapidly raised prices on life-saving drugs" by Follow, *Los Angeles Times* – https://www.latimes.com/business/la-fi-senate-drug-price-study-20161221-story.html

269. "Pluralistic: The public paid for "Moderna's" vaccine, and now we're going to pay again (and again and again); How Facebook's Real Names policy helps Cambodia's thin-skinned dictator terrorize dissenters (25 Jan 2023) – Pluralistic: Daily links from Cory Doctorow" by Cory Doctorow – https://pluralistic.net/2023/01/25/nationalize-moderna/

270. "Perfect Competition: Examples and How It Works" by Full Bio, *Investopedia* – https://www.investopedia.com/terms/p/perfectcompetition.asp

271. "Pluralistic: Look at all the great stuff we lost because of inflation scare-talk (05 May 2023) – Pluralistic: Daily links from Cory Doctorow" by Cory Doctorow – https://pluralistic.net/2023/05/05/wmds-two-point-oh/

272. "Bundling and Unbundling" by An Interview with Ben Thompson at the MoffettNathanson Media, Internet, and Communications Conference, *Stratechery by Ben Thompson* – https://stratechery.com/concept/business-models/bundling-and-unbundling/

273. "Crime Cartoon #207", *Andertoons* – https://andertoons.com/crime/cartoon/207/i-prefer-to-call-it-profit-sharing

274. *Anker* – https://www.anker.com/

275. "Blake Mycoskie Conceived The Idea For TOMS Shoes While Sitting On A Farm, Pondering Life, In Argentina" by SecondAct, *Business Insider* – https://www.businessinsider.com/blake-mycoskie-argentina-toms-shoes-2011-09

276. "TOMS Shoes Reviews - 1.4 Stars", *Sitejabber* – https://www.sitejabber.com/reviews/toms.com

277. https://twitter.com/patagonia/status/776223080779833344

278. https://twitter.com/sweatystartup/status/1621548157187284994?s=20

279. https://www.profitwell.com/recur/all/positive-onboarding-boosts-retention-wtp

280. "Pluralistic: How workers get trapped by "bondage fees"; Red Team Blues Chapter One, part five (21 Apr 2023) – Pluralistic: Daily links from Cory Doctorow" by Cory Doctorow – https://pluralistic.net/2023/04/21/bondage-fees/

281. https://twitter.com/doctorow/status/1628948906657878016?s=20

282. "Pluralistic: Venture predation (19 May 2023) – Pluralistic: Daily links from Cory Doctorow" by Cory Doctorow – https://pluralistic.net/2023/05/19/fake-it-till-you-make-it/

283. "Valentine Cartoon #9174", *Andertoons* – https://andertoons.com/valentine/cartoon/9174/hit-her-with-lower-expectations-first

284. "Valentines Cartoon #9173", *Andertoons* – https://andertoons.com/valentines/cartoon/9173/higher-next-time-hankering

285. "Michael Mauboussin - Sharpening Investor & Executive Toolkits - [Invest Like the Best, EP.308]", *Spotify* – https://open.spotify.com/episode/30smJfSR30QI1Nwk7YO130?go=1&sp_cid=210062e5df3dc0948de5cef2743a25d6

286. "Apple Net Profit Margin 2010-2025", *MacroTrends* – https://www.macrotrends.net/stocks/charts/AAPL/apple/net-profit-margin

287. "Manifesto for Agile Software Development" – https://agilemanifesto.org/

288. "The Complete Guide to the Kano Model", *Folding Burritos* – https://foldingburritos.com/blog/kano-model/

289. https://wiki.doing-projects.org/index.php/Kano_model

290. "Buffer: Social media management for everyone", *Buffer: All-you-need social media toolkit for small businesses* – https://buffer.com/

291. "Trump's Truth Social in trouble as financial, technical woes mount" by Drew Harwell, *The Washington Post* – https://www.washingtonpost.com/technology/2022/04/05/trump-truth-social-media-failure/

292. "What is", *Blue Ocean Strategy* — https://www.blueoceanstrategy.com/what-is-blue-ocean-strategy/
293. "English" — https://www.fairtrade.net/us-en.html
294. "Notifications: why less is more—how Facebook has been increasing both user satisfaction and app..." by Analytics at Meta, *Medium* — https://medium.com/@AnalyticsAtMeta/notifications-why-less-is-more-how-facebook-has-been-increasing-both-user-satisfaction-and-app-9463f7325e7d
295. https://twitter.com/dohertyjf
296. "News Cartoon #4033", *Andertoons* — https://andertoons.com/news/cartoon/4033/reaction-to-news-was-mixed-largely-because-we-asked-more-than-one-person
297. https://twitter.com/johncutlefish/status/1335822976957247489?s=21
298. "Legal Cartoon #1509", *Andertoons* — https://andertoons.com/legal/cartoon/1509/this-just-says-you-wont-reveal-anything-about-our-nondisclosure-agreement
299. "Meeting Cartoon #5327", *Andertoons* — https://andertoons.com/meeting/cartoon/5327/we-all-agree-lets-go-back-to-desks-discuss-why-this-wont-work
300. "R&D Spend - SaaS Benchmarks" by Alexanne Prosacco, *ScaleXP* — https://www.scalexp.com/blog/saas-benchmark/rd-spend/
301. "Let's Encrypt", *Let's Encrypt* — https://letsencrypt.org/
302. "Meeting Cartoon #512", *Andertoons* — https://andertoons.com/meeting/cartoon/512/so-as-you-can-see-customer-satisfaction-up-considerably-since-phasing-out-complaint-forms
303. https://twitter.com/maxlynch/status/1753066541795692547
304. "You're a little company, now act like one" by Jason Cohen, *A Smart Bear* — https://longform.asmartbear.com/little-company/
305. "Power law", *Wikipedia* — https://en.wikipedia.org/wiki/Power_law
306. "High availability", *Wikipedia* — https://en.wikipedia.org/wiki/High_availability
307. "Marriage Cartoon #6314", *Andertoons* — https://andertoons.com/marriage/cartoon/6314/kathy-if-you-agree-to-these-terms-of-service-click-i-do
308. "The Innovator's Dilemma", *Wikipedia* — https://en.wikipedia.org/wiki/The_Innovator%27s_Dilemma
309. *Twitter* — https://x.com/HarryStebbings/status/1766542812474978717?s=20
310. "Advertising Cartoon #106", *Andertoons* — https://andertoons.com/advertising/cartoon/106/its-not-that-i-dont-like-you-ted-you-just-dont-fit-my-target-demographic
311. "Pick one and own it" by Jason Cohen, *A Smart Bear* — https://longform.asmartbear.com/one-benefit/
312. "Ask Your Developer by Jeff Lawson, co-founder and CEO of Twilio" by Jeff Lawson — https://www.askyourdeveloper.com/
313. "How an Ad Campaign Made Lesbians Fall in Love with Subaru" by Alex Mayyasi, *Priceonomics* — https://priceonomics.com/how-an-ad-campaign-made-lesbians-fall-in-love-with/
314. https://knoema.com/infographics/floslle/top-vehicle-manufacturers-in-the-us-market-1961-2016
315. "Duck Cartoon #3613", *Andertoons* — https://andertoons.com/duck/cartoon/3613/all-im-saying-just-once-id-like-to-be-sitting-without-being-target
316. https://twitter.com/DanielZarick/status/1743010670763782204
317. https://twitter.com/dohertyjf/status/1742739614031323572
318. https://twitter.com/FlorianCaesar
319. "Chicken Cartoon #8796", *Andertoons* — https://andertoons.com/chicken/cartoon/8796/had-winner-winner-dream-again
320. *Pallyy* — https://pallyy.com/
321. https://twitter.com/Timb03/status/1697778180168593717?s=20

322. "Kit: Email-First Operating System for Creators (formerly ConvertKit)", *Kit* — https://kit.com/

323. *Twitter* — https://x.com/ayushtweetshere/status/1758714477007032710?s=20

324. "How ConvertKit Grew to $1.7M MRR with a Tiny Investment" by Tom Hunt, *Growjo* — https://growjo.com/blog/2020/06/17/how-convertkit-grew-to-1-7m-mrr-with-a-tiny-investment/

325. "How Nathan Barry Turned ConvertKit Into an $8M+ Business in 4 Years" by Alex Turnbull, *Groove Blog* — https://blog.groovehq.com/nathan-barry-interview

326. "Subscribe for a few deep articles per month" by Jason Cohen, *A Smart Bear* — https://longform.asmartbear.com/subscribe/

327. "WP Engine, the Smarter Way to Host WordPress", *WP Engine* — https://wpengine.com/

328. https://twitter.com/paulg/status/1642830071202283525?s=20

329. https://twitter.com/peldi

330. "Hit $100,000 in revenue, time to start looking up" by Peldi Guilizzoni — https://balsamiq.com/blog/hit-100000-in-revenue-time-to-start-looking-up/

331. https://twitter.com/dharmesh

332. "Holy Crap! HubSpot Has Now Raised A Total Of $33 Million" by Dharmesh Shah, *OnStartups* — https://www.onstartups.com/tabid/3339/bid/10799/holy-crap-hubspot-has-now-raised-a-total-of-33-million.aspx

333. "Investor Relations", *HubSpot* — https://ir.hubspot.com/

334. "Slack now has 1.1M daily active users, 300K paid seats, $25M in annual recurring revenue" by Jordan Novet, *VentureBeat* — https://venturebeat.com/business/slack-now-has-1-1m-daily-active-users-300k-paid-seats-25m-in-annual-recurring-revenue/

335. "Facebook to reach one billion users by end of summer" by Albizu Garcia, *The Sociable* — https://www.sociable.co/social-media/facebook-to-reach-one-billion-users-by-end-of-summer/

336. "Why The Craigslist Hack is Only AirBnb's Second Best Growth Hack" by Federico Etch.ai, *Medium* — https://medium.com/@etch.ai/how-airbnb-got-their-early-traction-cb059e902ea4

337. "11 Ways to Grow your YouTube Channel (June 2025)" by Trevor James, *The Food Ranger* — https://www.thefoodranger.com/how-to-grow-your-youtube-channel/

338. "Newsletter —" by 0, *Lenny Rachitsky* — https://www.lennyrachitsky.com/newsletter

339. *Twitter* — https://x.com/lennysan/status/1720485072758042749?s=20

340. https://twitter.com/aliatttar/status/1720490766210904113

341. *Twitter* — https://x.com/lennysan/status/1837158641968759274

342. "Buffer: Social media management for everyone", *Buffer: All-you-need social media toolkit for small businesses* — https://buffer.com/

343. "Transparent Metrics Dashboard", *Buffer: All-you-need social media toolkit for small businesses* — https://buffer.com/metrics

344. "SparkToro", *SparkToro Logo* — https://sparktoro.com/

345. "SparkToro Year 3 Retrospective: Investor Payback, Systemic Challenges, and V2 on the Way" by Rand Fishkin, *SparkToro* — https://sparktoro.com/blog/sparktoro-year-3-retrospective-investor-payback-systemic-challenges-and-v2-on-the-way/

346. https://twitter.com/MacBudkowski/status/1723724725392707706

347. "Sales Cartoon #6026", *Andertoons* — https://andertoons.com/sales/cartoon/6026/some-things-clicking-i-want-you-to-find-out-what-and-click-holy-hell-out-of-it

348. "More money if you do, more money if you don't" by Jason Cohen, *A Smart Bear* — https://longform.asmartbear.com/more-money/

349. "Out of the cesspool and into the sewer: A/B testing trap" by Jason Cohen, *A Smart Bear* — https://longform.asmartbear.com/local-minimum/

350. "Marketing Cartoon #7220", *Andertoons* — https://andertoons.com/marketing/cartoon/7220/i-miss-the-old-days-when-we-could-just-tout-things

351. *Twitter* – https://x.com/shaneharter/status/1459277283827412995
352. *Twitter* – https://x.com/pbteja1998/status/1615676112079835141
353. *Twitter* – https://x.com/aymanalabdul/status/1769715189295784092
354. *Twitter* – https://x.com/euboid/status/1835979554566893999
355. *Twitter* – https://x.com/Colleran/status/850447064664350723
356. *Twitter* – https://x.com/Patticus/status/1702313260547006942
357. "Money Cartoon #3130", *Andertoons* – https://andertoons.com/money/cartoon/3130/its-not-just-raise-in-my-allowance-its-also-boost-for-consumer-confidence
358. https://twitter.com/fabcaravita
359. "Profits Cartoon #7790", *Andertoons* – https://andertoons.com/profits/cartoon/7790/profits-excess-revenues-period-of-time-it-doesnt-look-good
360. "Sales Cartoon #7444", *Andertoons* – https://andertoons.com/sales/cartoon/7444/ok-lets-not-get-into-profit-shaming-here
361. "Schumpeter on Strategy" by Kirk Willard, *Reaction Wheel* – https://reactionwheel.net/2019/01/schumpeter-on-strategy.html
362. "12 Small Business Statistics: Facts & Numbers for [wcyear]" by Kazimierz Rajnerowicz, *Tidio* – https://www.tidio.com/blog/small-business-statistics/
363. "Ramen Profitable" – https://www.paulgraham.com/ramenprofitable.html
364. "Profit Cartoon #6129", *Andertoons* – https://andertoons.com/profit/cartoon/6129/and-we-all-lived-profitably-after
365. "Sales Cartoon #8781", *Andertoons* – https://andertoons.com/sales/cartoon/8781/pooh-way-pas-oh-bother
366. "BLUNDERS! Lecture with GM Ben Finegold", *YouTube* – https://www.youtube.com/watch?si=aVoR9dcmmvd6zgbZ&t=282&v=EDgRR7SGf0M&feature=youtu.be
367. "Elo rating system", *Wikipedia* – https://en.wikipedia.org/wiki/Elo_rating_system
368. "Avoiding Stupidity is Easier than Seeking Brilliance" by Shane, *Farnam Street* – https://fs.blog/avoiding-stupidity/
369. "Turtles all the way down", *Wikipedia* – https://en.wikipedia.org/wiki/Turtles_all_the_way_down
370. "Five whys", *Wikipedia* – https://en.wikipedia.org/wiki/Five_whys
371. "Black Swan Farming" – https://paulgraham.com/swan.html
372. "Graph Cartoon #7231", *Andertoons* – https://andertoons.com/graph/cartoon/7231/serendipity-is-up-fluke-doing-well-concerned-about-dumb-luck
373. *Amazon* – https://www.amazon.com/Zero-One-Notes-Startups-Future/dp/0804139296
374. "Thanksgiving Cartoon #4206", *Andertoons* – https://andertoons.com/thanksgiving/cartoon/4206/no-end-of-november-bad-for-me
375. "Crime Cartoon #8673", *Andertoons* – https://andertoons.com/crime/cartoon/8673/tell-us-what-you-know-want-big-one-russian-nesting-doll
376. "Valentine Cartoon #7163", *Andertoons* – https://andertoons.com/valentine/cartoon/7163/heart-arrows-not-for-everyone-side-effects-may-include-love
377. https://www.cbinsights.com/research/report/startup-failure-reasons-top/
378. "The 18 Mistakes That Kill Startups" – https://www.paulgraham.com/startupmistakes.html
379. "Why Start-ups Fail" by Tom Eisenmann, *Harvard Business Review* – https://hbr.org/2021/05/why-start-ups-fail
380. "Steve Blank 9 Deadliest Start-up Sins" by John S Wren, *Steve Blank* – https://steveblank.com/2012/05/14/9-deadliest-start-up-sins/
381. "business - Top 17 Startup Mistakes You Can't Afford to Make" by Mark Henricks, *Entrepreneur* – https://www.entrepreneur.com/business-news/business-top-17-startup-mistakes-you-can39t-afford-to/78474

382. "Why Startups Die" by TNW, *The Next Web* – https://thenextweb.com/news/why-startups-die

383. "Five Reasons 8 Out Of 10 Businesses Fail" by Eric T. Wagner, *Forbes* – https://www.forbes.com/sites/ericwagner/2013/09/12/five-reasons-8-out-of-10-businesses-fail/?sh=5d9302cd6978

384. "Change Cartoon #7548", *Andertoons* – https://andertoons.com/change/cartoon/7548/twelfth-final-thing-you-change-right-now-be-your-true-self

385. *Amazon* – https://www.amazon.com/Drive-Surprising-Truth-About-Motivates/dp/B0032COUMC/ref=sr_1_1?crid=19W217WIQZUM1&keywords=drive+daniel+pink&qid=1643834774&sprefix=boooks+the+ancient+cit%2Caps%2C90&sr=8-1

386. "The puzzle of motivation" by Dan Pink, *TED Talks* – https://www.ted.com/talks/dan_pink_the_puzzle_of_motivation?language=en

387. "Self-Determination Theory of Motivation - Center for Community Health & Prevention", *University of Rochester Medical Center* – https://www.urmc.rochester.edu/community-health/patient-care/self-determination-theory

388. "National Geographic", *Wikipedia* – https://en.wikipedia.org/wiki/National_Geographic

389. "Dan Buettner", *Wikipedia* – https://en.wikipedia.org/wiki/Dan_Buettner

390. "Ikigai", *Wikipedia* – https://en.wikipedia.org/wiki/Ikigai

391. *WPGraphQL* – https://www.wpgraphql.com/

392. https://twitter.com/jasonbahl

393. "The Golden Circle", *Simon Sinek* – https://simonsinek.com/golden-circle/

394. https://twitter.com/patagonia/status/776223080779833344

395. "Salesforce for Nonprofits", *Salesforce* – https://www.salesforce.com/nonprofit/

396. "Commited to Social Responsibility" by Somer Athari, Noble Studios2024 WP Engine APP Award Winner, *WP Engine* – https://wpengine.com/engine-for-good/

397. "Boiling frog", *Wikipedia* – https://en.wikipedia.org/wiki/Boiling_frog

398. "Ask HN: I don't want to be a founder anymore", *Hacker News* – https://news.ycombinator.com/item?id=14417758

399. "Why We Prefer Founding CEOs" by Ben Horowitz, *Andreessen Horowitz* – https://a16z.com/why-we-prefer-founding-ceos/

400. "Flow (psychology)", *Wikipedia* – https://en.wikipedia.org/wiki/Flow_%28psychology%29

401. https://twitter.com/asmartbear/status/1772302431994442196?s=46

402. "WP Engine, the Smarter Way to Host WordPress", *WP Engine* – https://wpengine.com/

403. https://twitter.com/heatherjbrunner

404. "Building Customer Driven SaaS Products" by Jason Evanish, *Building Customer Driven SaaS Products | Jason Evanish* – https://jasonevanish.com/

405. "Stock Cartoon #710", *Andertoons* – https://andertoons.com/stock/cartoon/710/stocks-rose-slightly-in-early-trading-then-plummeted-on-news-that-stocks-rose-slightly-in-early-trading

406. "Robot Cartoon #8802", *Andertoons* – https://andertoons.com/robot/cartoon/8802/uprising-calendar-friday

407. "Distribution vs. Innovation" by Alex Rampell, *Andreessen Horowitz* – https://a16z.com/distribution-vs-innovation/

408. "Disruptive Innovation – What is It and How Does It Work?" by Julia Kylliäinen, *Viima Solutions Oy* – https://www.viima.com/blog/disruptive-innovation

409. "Axel Springer and OpenAI license agreement is worth "tens of millions of euros" per year" by Matthias Bastian, *THE DECODER* – https://the-decoder.com/axel-springer-and-openai-license-agreement-is-worth-tens-of-millions-of-euros-per-year/

410. "Reddit Strikes $60M Deal to Train AI with User Posts" by Contxto – https://contxto.com/en/artificial-intelligence/reddit-strikes-60m-deal-to-train-ai-with-user-posts/

411. "Robot Cartoon #5518", *Andertoons* — https://andertoons.com/robot/cartoon/5518/two-robots-at-cocktail-party-have-nametags-showing-bar-codes-instead-of-names
412. https://www.mammothgrowth.com/blog/moving-upstream-how-to-avoid-common-mistakes-that-product-led-growth-companies-make-when-adding-a-sales-motion
413. "Hubspot Revenue Breakdown - FourWeekMBA" by Gennaro Cuofano, *What is The FourWeekMBA* — https://fourweekmba.com/hubspot-revenue-breakdown/
414. "HubSpot User and Revenue Stats (2025)" by Backlinko Team, *Backlinko* — https://backlinko.com/hubspot-users
415. "Inside Story: Behind HubSpot's $32 Million Investment From Salesforce, Google and Sequoia" by Brian Halligan, *HubSpot* — https://blog.hubspot.com/blog/tabid/6307/bid/10480/inside-story-behind-hubspot-s-32-million-investment-from-salesforce-google-and-sequoia.aspx
416. "If you're a SaaS entrepreneur, you need to study Hubspot" by Ryan Kaufman, *The Growth Playbook* — https://thegrowthplaybook.substack.com/p/if-youre-a-saas-entrepreneur-you
417. "Replacing the "SWOT" with The Landscape" by Jason Cohen, *A Smart Bear* — https://longform.asmartbear.com/swot/
418. "Personal: What's your story? Why start Balsamiq?" by Peldi Guilizzoni — https://balsamiq.com/blog/personal-whats-your-story-why-start-balsamiq/
419. "Balsamiq: Fast, focused wireframing tools" — https://balsamiq.com/
420. "Hit $100,000 in revenue, time to start looking up" by Peldi Guilizzoni — https://balsamiq.com/blog/hit-100000-in-revenue-time-to-start-looking-up/
421. https://twitter.com/nathanbarry
422. "Kit: Email-First Operating System for Creators (formerly ConvertKit)", *Kit* — https://kit.com/
423. *Twitter* — https://x.com/ayushtweetshere/status/1758714477007032710?s=20
424. https://twitter.com/Timb03
425. *Pallyy* — https://pallyy.com/
426. https://twitter.com/Timb03/status/1697778180168593717?s=20
427. "Economy Cartoon #3502", *Andertoons* — https://andertoons.com/economy/cartoon/3502/due-to-recent-economic-conditions-picture-worth-has-dropped-to-an-all-time-low-of-842-words
428. "File:Costco Wholesale logo 2010-10-26.svg", *Wikipedia* — https://en.m.wikipedia.org/wiki/File:Costco_Wholesale_logo_2010-10-26.svg
429. "File:Southwest Airlines logo 2014.svg", *Wikipedia* — https://en.m.wikipedia.org/wiki/File:Southwest_Airlines_logo_2014.svg
430. "File:Logo Vanguard.jpg - Wikimedia Commons", *Wikipedia* — https://commons.wikimedia.org/wiki/File:Logo_Vanguard.jpg
431. "File:Ikea logo.svg", *Wikipedia* — https://en.m.wikipedia.org/wiki/File:Ikea_logo.svg
432. "Survivorship bias", *Wikipedia* — https://en.wikipedia.org/wiki/Survivorship_bias
433. "Abraham Wald and the airplane diagram with red bullet holes – here's the origin story", *Cameron Moll* — https://cameronmoll.com/journal/abraham-wald-red-bullet-holes-origin-story
434. https://public.asmartbear.com/docs/AirplaneSurvivors1943.pdf
435. https://norvig.com/experiment-design.html
436. "Medical journal editors take hard line on drug research", *The Sydney Morning Herald* — https://www.smh.com.au/national/medical-journal-editors-take-hard-line-on-drug-research-20040910-gdjpq1.html
437. "Joseph Banks Rhine", *Wikipedia* — https://en.wikipedia.org/wiki/Joseph_Banks_Rhine
438. "Zener cards", *Wikipedia* — https://en.wikipedia.org/wiki/Zener_cards
439. "WP Engine, the Smarter Way to Host WordPress", *WP Engine* — https://wpengine.com/
440. "Steven Levitt", *Wikipedia* — https://en.wikipedia.org/wiki/Steven_Levitt

441. *Amazon* – https://www.amazon.com/gp/product/0061234001?ie=UTF8&tag=teamcohen-20&linkCode=as2&camp=1789&creative=9325&creativeASIN=0061234001

442. *Amazon* – https://www.amazon.com/gp/product/0066620996?ie=UTF8&tag=teamcohen-20&linkCode=as2&camp=1789&creative=9325&creativeASIN=0066620996

443. *Amazon* – https://www.amazon.com/gp/product/0060548789?ie=UTF8&tag=asmbe-20&linkCode=as2&camp=1789&creative=390957&creativeASIN=0060548789

444. "Accounting for Startups: Cash-basis or Accrual-basis?" by Jason, *@ASmartBear* – https://blog.asmartbear.com/cash-or-accrual-basis-accounting/

445. "Bending over: How to sell to large companies" by Jason, *@ASmartBear* – https://blog.asmartbear.com/selling-to-large-companies/

446. "June Huh, High School Dropout, Wins the Fields Medal" by Jordana Cepelewicz, *Quanta Magazine* – https://www.quantamagazine.org/june-huh-high-school-dropout-wins-the-fields-medal-20220705/

447. "Christmas Cartoon #4324", *Andertoons* – https://andertoons.com/Christmas/cartoon/4324/santa-sees-therapist-people-not-believing-in-him

448. "Impostor syndrome", *Wikipedia* – https://en.wikipedia.org/wiki/Impostor_syndrome

449. "The Impostor Syndrome" by Leigh Buchanan, *Inc.* – https://www.inc.com/magazine/20060901/handson-leadership.html

450. "Impostor phenomenon study: Most entrepreneurs affected" – https://kajabi.com/blog/impostor-phenomenon-study

451. "The Business Owner's "Imposter Complex"" by Posted by In Good Company WorkPlaces – https://ingoodcompanyworkplaces.blogspot.com/2009/11/business-owners-imposter-complex.html

452. https://sciencecareers.sciencemag.org/career_development/previous_issues/articles/2008_02_15/caredit_a0800025

453. https://twitter.com/search?q=%22impostor+syndrome%22&src=typed_query

454. "Psychiatry Cartoon #8014", *Andertoons* – https://andertoons.com/psychiatry/cartoon/8014/turns-out-my-imposter-syndrome-was-pretty-much-right-on

455. "Beating The Big Imposter", *The Mihir Chronicles* – https://mihirchronicles.com/beating-big-imposter/

456. "Neil Gaiman" by neil-gaiman, *Tumblr* – https://neil-gaiman.tumblr.com/post/160603396711/hi-i-read-that-youve-dealt-with-with-impostor

457. https://twitter.com/zencohen

458. "Scott W. Bradley - Facet Digital, LLC" by Scott W. Bradley, *LinkedIn* – https://www.linkedin.com/in/scottwb69/

459. "File:Humboldt 1810 pp 47 48 50 51 52.jpg - Wikimedia Commons", *Wikipedia* – https://commons.wikimedia.org/wiki/File:Humboldt_1810_pp_47_48_50_51_52.jpg

460. "Dresden Codex", *Wikipedia* – https://en.wikipedia.org/wiki/Dresden_Codex

461. "Boris Podolsky", *Wikipedia* – https://en.wikipedia.org/wiki/Boris_Podolsky

462. "Nathan Rosen", *Wikipedia* – https://en.wikipedia.org/wiki/Nathan_Rosen

463. "Einstein–Podolsky–Rosen paradox", *Wikipedia* – https://en.wikipedia.org/wiki/EPR_paradox

464. "Erwin Schrödinger" by DA00036675, *Wikidata* – https://www.wikidata.org/wiki/Q9130

465. "Schrödinger equation", *Wikipedia* – https://en.wikipedia.org/wiki/Schr%C3%B6dinger_equation

466. "Schrödinger's cat", *Wikipedia* – https://en.wikipedia.org/wiki/Schr%C3%B6dinger%27s_cat

467. "Quantum Mechanics, The Uncertainty Principle, Schrodinger's Cat, & Other Misconceptions –" by Jon Therkildsen, *MOVIESANDSCIENCE.COM* – https://moviesandscience.com/blog/science/quantum-mechanics

468. "The Periodic Table of Content Marketing is Finally Out!" by Susan Serena, *LinkedIn* – https://www.linkedin.com/pulse/periodic-table-content-marketing-finally-out-susan-douglin/

469. "Doctrine" by swardley, *wardleymaps* – https://medium.com/wardleymaps/doctrine-8bb0015688e5

470. "Good Strategy/Bad Strategy & Playing to Win" by Roger Martin, *Medium* – https://rogermartin.medium.com/good-strategy-bad-strategy-playing-to-win-b7b415afbd43

471. "Selling Cartoon #8671", *Andertoons* – https://andertoons.com/selling/cartoon/8671/executives-who-liked-this-graph-also-liked

472. "11 Real-Life Insider Threat Examples" by mimecast, *Mimecast* – https://www.mimecast.com/blog/insider-threat-examples/

473. "Elon Used 'Simple' Email Trick To Catch Employee Leaking Tesla Data To The Press" by Vinay Patel, *International Business Times UK* – https://www.ibtimes.co.uk/elon-used-simple-email-trick-catch-employee-leaking-tesla-data-press-1724419

474. "Car Cartoon #3220", *Andertoons* – https://andertoons.com/car/cartoon/3220/i-like-it-but-im-looking-for-more-of-status-symbol-any-way-you-can-double-price

475. *Twitter* – https://x.com/alxberman/status/1914328520861286705?s=12

476. "Business Cartoon #6515", *Andertoons* – https://andertoons.com/business/cartoon/6515/strategy-big-data-synergized-paradigms-something-that-rhymes

477. https://www.tesla.com/blog/secret-tesla-motors-master-plan-just-between-you-and-me

478. https://twitter.com/chamath/status/1369328384812093441

479. "The Anatomy of an Amazon 6-pager" by Jesse Freeman, *The Writing Cooperative* – https://writingcooperative.com/the-anatomy-of-an-amazon-6-pager-fc79f31a41c9?gi=4f9623d665c8

480. "Create Strategic Company Alignment With a V2MOM" by Marc Benioff December 11, 2024 6 min read, *Salesforce* – https://www.salesforce.com/blog/how-to-create-alignment-within-your-company/

481. https://twitter.com/shreyas/status/1384008853826592776

482. "Lean Canvas" – https://www.leanfoundry.com/tools/lean-canvas

483. "The Secret Tesla Motors Master Plan (just between you and me)", *Tesla* – https://www.tesla.com/secret-master-plan

484. "The Midwit Trap" by Philo, *MD&A* – https://www.md-a.co/p/the-midwit-trap

485. "Greatness Starts with Tying Your Shoes" by Kirk j Barbera, *Ascent Publication* – https://medium.com/the-ascent/greatness-starts-with-tying-your-shoes-59fbeff027dc

486. https://twitter.com/TheCoachJournal/status/1171955834462646272

487. "Kid Cartoon #4501", *Andertoons* – https://andertoons.com/kid/cartoon/4501/lets-say-honesty-isnt-going-to-work-whats-second-best-policy

488. "Strategy", *Thinker Author Advisor Speaker* – https://rogerlmartin.com/thought-pillars/strategy

489. "Pick one and own it" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/one-benefit/

490. "Eigenquestions: The Art of Framing Problems" by Shishir Mehrotra, *Coda* – https://coda.io/@shishir/eigenquestions-the-art-of-framing-problems

491. "Voting Cartoon #8066", *Andertoons* – https://andertoons.com/voting/cartoon/8066/rest-assured-however-i-vote-it-will-be-staunchly

492. "craigslist: austin jobs, apartments, for sale, services, community, and events", *craigslist* – https://austin.craigslist.org/

493. https://smallbusiness.chron.com/craigslist-money-27287.html

494. "Craigslist: The remarkable but potentially obsolete success story of the "ultimate platform"" by Eve, *Digital Innovation and Transformation* – https://d3.harvard.edu/platform-digit/submission/craigslist-the-remarkable-but-potentially-obsolete-success-story-of-the-ultimate-platform/

495. "Here's Proof Every Startup Idea You've Ever Had Craigslist Already Does" by Alyson Shontell, *Business Insider* – https://www.businessinsider.com/craigslist-versus-startups-chart-2014-8

496. "As Vertical Marketplaces Rise, Craigslist Faces Its Demise" by Jeremy Levine, Rafi Syed, *TechCrunch* – https://techcrunch.com/2015/04/10/as-vertical-marketplaces-rise-craigslist-faces-its-demise/

497. "BURNING SENSATION" by Adam Lashinsky, *December 12, 2005* – https://money.cnn.com/magazines/fortune/fortune_archive/2005/12/12/8363113/index.htm

498. "Top 100: The Most Visited Websites in the US [2025 Top Websites Edition]", *Semrush* – https://www.semrush.com/blog/most-visited-websites/

499. "Brighton hotel bombing", *Wikipedia* – https://en.wikipedia.org/wiki/Brighton_hotel_bombing

500. "Unpacking growth with Andy Johns, Partner at Unusual Ventures" by Erasmus Elsner, CFA, *Sand Hill Road* – https://medium.com/sand-hill-road/unpacking-growth-with-andy-johns-partner-at-unusual-ventures-79778fc7bdd8

501. "The Twenty Minute VC (20VC): Venture Capital", *The Pitch: 20Growth: Top Growth Lessons from the Early Facebook, Twitter and Quora Days, The Most Important Question to Ask When Building Your Growth Team, How To Test For True Candidate Depth and Quality When Hiring for Growth with Andy Johns, Venture Partner @ Unu* – https://thetwentyminutevc.libsyn.com/20growth-top-growth-lessons-from-the-early-facebook-twitter-and-quora-days-the-most-important-question-to-ask-when-building-your-growth-team-how-to-test-for-true-candidate-depth-and-quality-when-hiring-for-growth-with-andy-johns-venture-partner-unusual-ve

502. "Amazon Confirms It Makes No Profit On Kindles" by Kelly Clay, *Forbes* – https://www.forbes.com/sites/kellyclay/2012/10/12/amazon-confirms-it-makes-no-profit-on-kindles/?sh=3f08d58d6b43

503. "4 Reasons Amazon's Fire Phone Was a Flop" by Victor Luckerson, *Time* – https://time.com/3536969/amazon-fire-phone-bust/

504. "Jeff Bezos: I've made billions of dollars of failures at Amazon" by Samuel Gibbs, *The Guardian* – https://www.theguardian.com/technology/2014/dec/03/jeff-bezos-billions-dollars-failures-amazon

505. "Singing the Data Analytics Blues" by Roger Martin, *Medium* – https://rogermartin.medium.com/singing-the-data-analytics-blues-58fdf93b27bf

506. "Time management", *Wikipedia* – https://en.wikipedia.org/wiki/Time_management

507. "Work Cartoon #2714", *Andertoons* – https://andertoons.com/work/cartoon/2714/youre-right-its-not-much-of-mission-statement

508. "The Magical Number Seven, Plus or Minus Two", *Wikipedia* – https://en.wikipedia.org/wiki/The_Magical_Number_Seven,_Plus_or_Minus_Two

509. "WTF is Strategy?" by Vince Law, *HackerNoon* – https://hackernoon.com/wtf-is-a-strategy-bcaa3fda9a31

510. *Amazon* – https://www.amazon.com/Good-Strategy-Bad-Difference-Matters/dp/0307886239

511. "Notes from "Good Strategy / Bad Strategy" by Jeff Zych" by Jeff Zych, *Jeff Zych's Internet Nook* – https://jlzych.com/2018/06/27/notes-from-good-strategy-bad-strategy/

512. "What is", *Blue Ocean Strategy* – https://www.blueoceanstrategy.com/what-is-blue-ocean-strategy/

513. "How to Define Your Product Strategy" by Gibson Biddle, *Medium* – https://gibsonbiddle.medium.com/intro-to-product-strategy-60bdf72b17e3

514. "What is Optionality?" by Taylor, *Taylor Pearson* – https://taylorpearson.me/optionality/

515. https://onlinelibrary.wiley.com/doi/pdf/10.1002/smj.4250110302

516. "Sign Cartoon #4019", *Andertoons* – https://andertoons.com/sign/cartoon/4019/were-not-changing-anything-we-just-wanted-to-charge-more

517. "Wine Cartoon #305", *Andertoons* – https://andertoons.com/wine/cartoon/305/and-for-connoisseur-on-budget-we-have-lovely-tuesday-vintage-at-buck-fifty

518. "The mid-market briar patch" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/mid-market/

519. "Car Cartoon #3220", *Andertoons* – https://andertoons.com/car/cartoon/3220/i-like-it-but-im-looking-for-more-of-status-symbol-any-way-you-can-double-price

520. *Amazon* – https://www.amazon.com/s?k=crossing+the+chasm+geoffrey+moore&hvadid=580635163611&hvdev=c&hvlocphy=9028297&hvnetw=g&hvqmt=e&hvrand=3561795095205753345&hvtargid=kwd-4993426152&hydadcr=22594_13493210&tag=googhydr-20&ref=pd_sl_16btg2x8z_e

521. ""ROI" is the wrong way to sell your product" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/roi-selling/

522. "Act like your price just doubled" by Jason, *@ASmartBear* – https://blog.asmartbear.com/act-like-your-price-just-doubled/

523. "Sales Cartoon #6927", *Andertoons* – https://andertoons.com/sales/cartoon/6927/things-good-but-avoid-pride-remember-what-happened-last-time

524. "Balancing Exploration and Exploitation" by Roger Martin, *Medium* – https://rogermartin.medium.com/balancing-exploration-and-exploitation-adb82146837?ref=anand-panicker.com

525. "8 Principles of The Innovator's Solution" by Tim Woods, *HYPE Innovation* – https://www.hypeinnovation.com/blog/8-principles-of-the-innovators-solution

526. https://www.jstor.org/stable/2634940

527. "Deep, Dark Thoughts On Being Different" by Dharmesh Shah, *LinkedIn* – https://www.linkedin.com/pulse/deep-dark-thoughts-being-different-dharmesh-shah/

528. "Pick one and own it" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/one-benefit/

529. "Detective Cartoon #8509", *Andertoons* – https://andertoons.com/detective/cartoon/8509/at-first-solve-crime-now-making-afghan

530. "Strong opinions, somewhat weakly held" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/strong-opinions-weakly-held/

531. "Software Quality Mortgage" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/software-quality-mortgage/

532. "Egg Cartoon #7238", *Andertoons* – https://andertoons.com/egg/cartoon/7238/i-put-all-the-eggs-in-the-same-basket-just-to-feel-alive

533. "Case Study: Using an LOI to get customer feedback on a minimum viable product" – https://www.startuplessonslearned.com/2009/10/case-study-using-loi-to-get-customer.html

534. https://evhead.com/2005/11/ten-rules-for-web-startups.asp

535. "About the Cranky Product Manager" by Sue Raisty, *The Blog of Sue Raisty* – https://blog.sueraisty.com/about-crankypm/

536. "Interview Cartoon #7470", *Andertoons* – https://andertoons.com/interview/cartoon/7470/came-up-brilliant-idea-revolutionized-industry-saved-ruin

537. "Puzzle Cartoon #7891", *Andertoons* – https://andertoons.com/puzzle/cartoon/7891/stop-calling-it-a-head-scratcher-i-know-its-head-scratcher

538. "Painstik" by Contributors to Memory Alpha, *Fandom, Inc.* – https://memory-alpha.fandom.com/wiki/Painstik

539. "Shoshin", *Wikipedia* – https://en.wikipedia.org/wiki/Shoshin

540. "Sales Cartoon #8465", *Andertoons* – https://andertoons.com/sales/cartoon/8465/dunno-what-if-we-leveraged-something

541. "Twitter is over capacity." by Remko van Dokkum, *Flickr* – https://www.flickr.com/photos/remkovandokkum/5151041232

542. "File:Mark Zuckerberg - Move Fast and Break Things.jpg - Wikimedia Commons", *Wikipedia* – https://commons.wikimedia.org/wiki/File:Mark_Zuckerberg_-_Move_Fast_and_Break_Things.jpg

543. "Move Fast and Break Things" by Jonathan Taplin, *Audible.com* – https://www.audible.com/pd/Move-Fast-and-Break-Things-Audiobook/B071NFMZZQ

544. "File:Mark Zuckerberg - Move Fast with Stable Infra.jpg - Wikimedia Commons", *Wikipedia* – https://commons.wikimedia.org/wiki/File:Mark_Zuckerberg_-_Move_Fast_with_Stable_Infra.jpg

545. "Superman Cartoon #8360", *Andertoons* – https://andertoons.com/superman/cartoon/8360/superman-psychiatry-upset-cant-see-through-lead

546. "Google SRE - Continuous Improvement To Get Reliability" by Steven Thurgood and David Ferguson with Alex Hidalgo and Betsy Beyer, *Google SRE* – https://sre.google/workbook/implementing-slos/

547. "Revenue-Driven SEO Agency", *Ebarque logo* – https://www.embarque.io/

548. https://db-engines.com/en/ranking_trend/system/MySQL%3BPostgreSQL

549. "Peter Thiel (Ft. Blake Masters) – Zero to One Chapter 8: Secrets", *Genius* – https://genius.com/Peter-thiel-zero-to-one-chapter-8-secrets-annotated

550. "The Innovator's Dilemma", *Wikipedia* – https://en.wikipedia.org/wiki/The_Innovator%27s_Dilemma

551. "Microsoft targets Google's search dominance with AI-powered Bing", *Financial Times* – https://www.ft.com/content/2d48d982-80b2-49f3-8a83-f5afef98e8eb

552. "What is", *Blue Ocean Strategy* – https://www.blueoceanstrategy.com/what-is-blue-ocean-strategy/

553. "What Is Strategy?" by Michael E. Porter, *Harvard Business Review* – https://hbr.org/1996/11/what-is-strategy

554. "Strategy", *Thinker Author Advisor Speaker* – https://rogerlmartin.com/thought-pillars/strategy

555. "Alternative to the MVP (minimum viable product): Simple, Lovable and Complete (SLC)" by ANANT KAJALE, *LinkedIn* – https://www.linkedin.com/pulse/alternative-mvp-minimum-viable-product-simple-lovable-anant-kajale/

556. https://www.one80training.com/slc-vs-mvp-elevating-your-product-strategy/

557. "SLC vs. MVP: What's the difference, and which is best for your software project? –" by David Andreu, *Basecode* – https://basecode.ca/blog/slc-vs-mvp-whats-the-difference-and-which-is-best-for-your-software-project

558. "Simple, Lovable, Complete (SLC) Products: A Comprehensive Overview" by Anshad Ameenza, *Hubtree Ventures* – https://www.hubtree.ventures/post/simple-lovable-complete-slc-products-a-comprehensive-overview

559. "Inside Product: Introduction to Simple, Lovable and Complete (SLC) Prototyping" by Nimay Parekh, *Medium* – https://medium.com/@nimay/shifting-prototyping-mindsets-from-mvp-to-slc-eab5bc42b6b2

560. "Why SLC may be better than MVP for project development", *We Create Digital* – https://wecreate.digital/blog/mvp-vs-slc/

561. "What is a Minimum Loveable Product?" by Ziemowit Kaczmarski, *10Clouds* – https://10clouds.com/blog/design/minimum-loveable-product-what-is-it/

562. *Twitter* – https://x.com/s4nxbt/status/1918946145780568501

563. "Difference Between Minimum Viable Product (MVP) & SLC" by bluentspy, *BluEnt Technology* – https://www.bluent.net/blog/mvp-vs-slc/

564. "Creating a Simple, Lovable, and Complete Product: A Guide to SLC", *Helio* – https://helio.app/product-discovery/minimum-viable-product-alternatives/slc/

565. "MVP Alternative: Strategy, Types, And Examples" by Earteza Auvee, *Impala Intech* – https://impalaintech.com/blog/mvp/alternative-to-mvp-minimum-viable-product/

566. https://codelation.com/explaining-a-minimum-viable-product/

567. *Twitter* – https://x.com/Kindfiree/status/1936039417912938755

568. "Daugherty ProTip: Simple, Lovable, Complete", *YouTube* – https://www.youtube.com/watch?si=klb0BWBnriOBkZZe&v=2F0uSW8j72I&feature=youtu.be

569. "Subscribe for a few deep articles per month" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/subscribe/

570. "Hubble's Deep Fields" by Elizabeth Tammi, *NASA Science* – https://science.nasa.gov/mission/hubble/science/universe-uncovered/hubble-deep-fields/

571. "Tattoo Cartoon #5340", *Andertoons* – https://andertoons.com/tattoo/cartoon/5340/this-oh-please-its-not-even-mine

572. "StrengthsFinder 2.0" by Gallup, Inc., *Gallup* – https://www.gallup.com/cliftonstrengths/en/254033/strengthsfinder.aspx

573. "The true meaning of common idioms" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/idioms/

574. "CliftonStrengths 34" by Gallup, *Gallup Store* – https://store.gallup.com/p/en-us/10003/cliftonstrengths-34?c=1

575. "Vision Cartoon #6668", *Andertoons* – https://andertoons.com/vision/cartoon/6668/my-greatest-weakness-id-have-to-say-depth-perception

576. "Replacing the "SWOT" with The Landscape" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/swot/

577. "George Orwell: Why I Write" by O. Dag – https://orwell.ru/library/essays/wiw/english/e_wiw

578. "Bee Cartoon #8803", *Andertoons* – https://andertoons.com/bee/cartoon/8803/funeral-stinging-random-guy

579. "Brainstorming Doesn't Work; Try This Technique Instead" by Rebecca Greenfield, *Fast Company* – https://www.fastcompany.com/3033567/brainstorming-doesnt-work-try-this-technique-instead

580. *Amazon* – https://www.amazon.com/Creative-Conspiracy-Rules-Breakthrough-Collaboration/dp/1422173348?tag=wwwfccom-20

581. "Virtual communication curbs creative idea generation" by Melanie S. Brucks, *Nature* – https://www.nature.com/articles/s41586-022-04643-y?error=cookies_not_supported&code=05ff63b6-7914-4ff4-9536-debf15ddf826

582. "Look what we did" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/proud-of-founding/

583. "Software Quality Mortgage" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/software-quality-mortgage/

584. "Pitch Cartoon #7856", *Andertoons* – https://andertoons.com/pitch/cartoon/7856/this-about-wheel-again-og-because-if-it-is-i-swear-to-god

585. "Autobahn Consultants - Strategic, Recession and Growth Planning", *Autobahn Consultants* – https://autobahnconsultants.com/

586. "Pick one and own it" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/one-benefit/

587. "Laws of 10x found everywhere" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/10x/

588. "Constraints", *Cynefin.io* – https://cynefin.io/wiki/Constraints

589. "Design Under Constraint: How Limits Boost Creativity" by WIRED Staff, *WIRED* – https://www.wired.com/2009/02/dp-intro/

590. "The Psychology of Limitation: How Constraints Make Us More Creative" by Belle Beth Cooper, *Buffer: All-you-need social media toolkit for small businesses* – https://buffer.com/resources/7-examples-of-how-creative-constraints-can-lead-to-amazing-work/

591. "Art Cartoon #7178", *Andertoons* – https://andertoons.com/art/cartoon/7178/piece-touching-themes-joy-desire-loss-fourth-quarter-earnings

592. "Paradox Cartoon #7513", *Andertoons* – https://andertoons.com/paradox/cartoon/7513/schrodingers-shop-is-both-open-and-closed

593. "The mid-market briar patch" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/mid-market/

594. "Advice Cartoon #8342", *Andertoons* — https://andertoons.com/advice/cartoon/8342/devil-angel-live-a-little

595. "Home", *Thinker Author Advisor Speaker* — https://rogerlmartin.com/

596. "What Is Strategy?" by Michael E. Porter, *Harvard Business Review* — https://hbr.org/1996/11/what-is-strategy

597. *Amazon* — https://www.amazon.com/Blue-Ocean-Strategy-Expanded-Uncontested/dp/1625274491/ref=asc_df_1625274491/?tag=hyprod-20&linkCode=df0&hvadid=693465227026&hvpos=&hvnetw=g&hvrand=15655197287935843831&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9028297&hvtargid=pla-318700082110&psc=1&mcid=56178674db62348fb13db63425b6f6f4&gad_source=1&gclid=Cj0KCQjwltKxBhDMARIsAG8KnqUlPAscTCb9Jh-SJSICR6vc-0YuAIfc7NFMetccl9l0lX9XFgrjjx0YaAmNuEALw_wcB

598. "Decision Cartoon #6618", *Andertoons* — https://andertoons.com/decision/cartoon/6618/do-it-dont-do-it-we-dont-care-anymore

599. https://twitter.com/awazels

600. "The AI workspace that works for you.", *Notion* — https://adamwaselnuk.notion.site/Binstack-Make-tough-decisions-b3a80f7c06a44d77b180d7bd252bbfb1

601. "Voting Cartoon #8066", *Andertoons* — https://andertoons.com/voting/cartoon/8066/rest-assured-however-i-vote-it-will-be-staunchly

602. "Pencil Cartoon #8950", *Andertoons* — https://andertoons.com/pencil/cartoon/8950/golf-pencil-no-eraser-decisive

603. https://twitter.com/levie/status/1313236118117076993?s=12

604. "Animal Cartoon #5934", *Andertoons* — https://andertoons.com/animal/cartoon/5934/i-know-hes-not-much-to-look-at-but-symbiosis-great

605. "WP Engine, the Smarter Way to Host WordPress", *WP Engine* — https://wpengine.com/

606. "SmartBear Software - The confidence behind your code.", *SmartBear Software* — https://smartbear.com/

607. "The leading provider of meaningless marketing solutions" by Jason Cohen, *A Smart Bear* — https://longform.asmartbear.com/the-leading-provider/

608. "Bed, Bath, Linens, Things, Beyond, and More!" by Jason Cohen, *A Smart Bear* — https://longform.asmartbear.com/and-more/

609. "Is it OK to Sucks?" by Jason Cohen, *A Smart Bear* — https://longform.asmartbear.com/is-it-ok-to-sucks/

610. "Gapingvoid" — https://www.gapingvoid.com/

611. "Outside-In" by Jason Cohen, *A Smart Bear* — https://longform.asmartbear.com/outside-in/

612. "Cosmological argument", *Wikipedia* — https://en.wikipedia.org/wiki/Cosmological_argument

613. "Language Cartoon #7291", *Andertoons* — https://andertoons.com/language/cartoon/7291/buzzword-jargon-hyperbole-trite-rhyming-platitude-looks-good

614. "Hello, I'm 1074018628" by Jason Cohen, *A Smart Bear* — https://longform.asmartbear.com/hello-support/

615. "You're a little company, now act like one" by Jason Cohen, *A Smart Bear* — https://longform.asmartbear.com/little-company/

616. "Women Cartoon #6034", *Andertoons* — https://andertoons.com/women/cartoon/6034/ugh-i-could-feel-them-monetizing-me-with-their-eyes

617. "Software Quality Mortgage" by Jason Cohen, *A Smart Bear* — https://longform.asmartbear.com/software-quality-mortgage/

618. *Rob Walling* — https://robwalling.com/

619. *Twitter* — https://x.com/robwalling/status/1823345489560944868

620. *Twitter* — https://x.com/daniel_nguyenx/status/1823349536414445893

621. *Twitter* — https://x.com/NafetsWirth/status/1823417820170051624

622. *Twitter* — https://x.com/simonvlc

623. "El código es tu enemigo" by Simón Muñoz, *Estrategia de Producto* — https://www.estrategiadeproducto.com/p/el-codigo-es-tu-enemigo
624. "Work Cartoon #6720", *Andertoons* — https://andertoons.com/work/cartoon/6720/you-redefine-key-metrics-dynamic-optimization-alignment-you-figure-what-that-means
625. "Current reality tree (theory of constraints)", *Wikipedia* — https://en.wikipedia.org/wiki/Current_reality_tree_(theory_of_constraints)
626. "Every Product Needs a North Star Metric: Here's How to Find Yours" by Julia Dillon, *Amplitude* — https://amplitude.com/blog/product-north-star-metric
627. "Choosing Your North Star Metric" by Lenny Rachitsky, *Future* — https://future.com/north-star-metrics/
628. "Objectives and key results", *Wikipedia* — https://en.wikipedia.org/wiki/OKR
629. "WP Engine, the Smarter Way to Host WordPress", *WP Engine* — https://wpengine.com/
630. "Google SRE - Continuous Improvement To Get Reliability" by Steven Thurgood and David Ferguson with Alex Hidalgo and Betsy Beyer, *Google SRE* — https://sre.google/workbook/implementing-slos/
631. "Your Startup is a System You Can Map to Identify Problems, Align the Team & Win" by Ben Yoskovitz, *Focused Chaos* — https://www.focusedchaos.co/p/your-startup-is-a-system-you-can-map
632. "Set—and achieve—SMART-er goals [2025]" by Julia Martins, *Asana* — https://asana.com/resources/smart-goals
633. "With Goals, FAST Beats SMART" by Donald Sull and Charles Sull, *MIT Sloan Management Review* — https://sloanreview.mit.edu/article/with-goals-fast-beats-smart/
634. "North Star Playbook", *Amplitude* — https://amplitude.com/books/north-star
635. "Jaakko Piipponen" by Jaakko Piipponen, *Medium* — https://jaakkopiipponen.com/?gi=de5dd39a132a
636. https://twitter.com/JonathanDrake
637. https://twitter.com/vinodv
638. "Psychiatry Cartoon #8014", *Andertoons* — https://andertoons.com/psychiatry/cartoon/8014/turns-out-my-imposter-syndrome-was-pretty-much-right-on
639. "Radical Candor Book: Business Leadership Book For Better Bosses" by Razorfrog Web Design, *Radical Candor* — https://www.radicalcandor.com/the-book/
640. *Amazon* — https://www.amazon.com/Five-Dysfunctions-Team-Leadership-Fable/dp/0787960756
641. "How to overcome the 5 Dysfunctions of Virtual Teams - Tomorrow's Leadership" by Tomorrow's Leadership, *Tomorrow's Leadership* — https://tomorrowsleadership.nl/how-to-overcome-the-5-dysfunctions-of-virtual-teams/
642. *Amazon* — https://www.amazon.com/Good-Strategy-Bad-Difference-Matters/dp/0307886239
643. *Amazon* — https://www.amazon.com/Good-Great-Some-Companies-Others/dp/0066620996
644. "Chart Cartoon #3709", *Andertoons* — https://andertoons.com/chart/cartoon/3709/things-went-from-bad-to-worse-but-were-hopeful-now-that-were-doing-badly-again
645. "Drake equation", *Wikipedia* — https://en.wikipedia.org/wiki/Drake_equation
646. "Frank Drake", *Wikipedia* — https://en.wikipedia.org/wiki/Frank_Drake
647. "The leading provider of meaningless marketing solutions" by Jason Cohen, *A Smart Bear* — https://longform.asmartbear.com/the-leading-provider/
648. https://twitter.com/asmartbear/status/1820192235981332735
649. "Smartphone Cartoon #8362", *Andertoons* — https://andertoons.com/smartphone/cartoon/8362/hey-said-take-us-to-leader-hello-aliens
650. "Hospital Cartoon #7349", *Andertoons* — https://andertoons.com/hospital/cartoon/7349/rejection-rate-organ-transplant-high-cram-three-or-four-in

651. "Reddit - The heart of the internet", *Log In* – https://www.reddit.com/r/Design/comments/3latn6/if_engineers_were_treated_like_designers/

652. "Darwinian advice for "Going Viral"" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/going-viral/

653. "Jeff Bezos: I've made billions of dollars of failures at Amazon" by Samuel Gibbs, *The Guardian* – https://www.theguardian.com/technology/2014/dec/03/jeff-bezos-billions-dollars-failures-amazon?utm_source=longform.asmartbear.com&utm_campaign=longform.asmartbear.com&utm_medium=post

654. *Risk First* – https://riskfirst.org/

655. "Aliens Cartoon #7878", *Andertoons* – https://andertoons.com/aliens/cartoon/7878/well-probe-you-but-first-id-like-to-get-to-know-real-you

656. "Dating Cartoon #8176", *Andertoons* – https://andertoons.com/dating/cartoon/8176/super-dark-bachelor-number-three-have-to-admit-im-intrigued

657. "You're a little company, now act like one" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/little-company/

658. "Home", *Noah Kagan* – https://noahkagan.com/

659. "Marketing 101: Learn from Noah Kagan of Sumo & Okdork" by David Fallarme, *The Marketing Student* – https://www.themarketingstudent.com/noah-kagan-launch/

660. "SOP: engage Slack communities" – https://docs.google.com/document/d/e/2PACX-1vQM5EufRTuKeJ_q11vMSTBDyXzRkf0sYSyVVmCmkSEFfj-EK93E0DP8BsQZAxQ4i5U2K_etvSH6kseri/pub

661. "Hacker News" – https://news.ycombinator.com/

662. *Quora* – https://www.quora.com/

663. "craigslist > sites", *craigslist®* – https://www.craigslist.org/about/sites

664. *Twitter* – https://x.com/gridpane/status/1868687475784216948

665. "SOP: using review sites to get calls booked" – https://docs.google.com/document/d/e/2PACX-1vSe8-jZ4wNPSF9jdRQAgaMUmM-Os4kCpkhpoltpFLIM_CE-gsAmZIyZTVS7fFN2Y68GZ-DYTImS7jg4/pub

666. *Twitter* – https://x.com/JohnPaulAndersn/status/1868578197992005923

667. "@martyamark.bsky.social", *Bluesky Social* – https://bsky.app/profile/martyamark.bsky.social/post/3ldghyc7bl22p

668. "User Interviews" by See Report →, *User Interviews logo with rainbow filled icon* – https://www.userinterviews.com/

669. https://www.fiverr.com/

670. https://www.upwork.com/

671. "Homepage", *GLG* – https://glginsights.com/

672. "Homepage", *Third Bridge* – https://thirdbridge.com/

673. "HubSpot Blog" by Maxwell Iskiev, *HubSpot* – https://blog.hubspot.com/

674. *Amazon* – https://www.amazon.com/Inbound-Marketing-Revised-Updated-Customers/dp/1118896653

675. "Nick Basile (@nick-basile.com)", *Bluesky Social* – https://bsky.app/profile/nick-basile.com/post/3ldggbwrkrc2i

676. "Marketing at early startups: Deep, not wide" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/one-channel/

677. https://www.talkingtohumans.com/

678. *Meetup* – https://www.meetup.com/

679. https://www.momtestbook.com/

680. "Deploy Empathy Home - Deploy Empathy", *Deploy Empathy - A practical guide to interviewing customers* – https://deployempathy.com/

681. *Amazon* – https://www.amazon.com/s?k=traction+gabriel+weinberg&hvadid=580712098231&hvdev=c&hvlocphy=9207466&hvnetw=g&hvqmt=e&

hvrand=13366269211832681115&hvtargid=kwd-91263434250&hydadcr=21905_13324233&tag=googhydr-20&ref=pd_sl_dvd6clqld_e

682. "Carter Bryden 🅒🅐 (@carterbryden.com)", *Bluesky Social* – https://bsky.app/profile/carterbryden.com

683. *Counterplot* – https://counterplot.co/

684. "giffconstable.com", *Thoughts on startups, product, innovation and more, by Giff Constable* – https://giffconstable.com/

685. "Request for ideas!" by Jason Cohen, *LinkedIn* – https://www.linkedin.com/feed/update/urn:li:activity:7274339476414312448?commentUrn=urn%3Ali%3Acomment%3A%28activity%3A7274339476414312448%2C7274342624478580736%29&dashCommentUrn=urn%3Ali%3Afsd_comment%3A%287274342624478580736%2Curn%3Ali%3Aactivity%3A7274339476414312448%29

686. *Twitter* – https://x.com/kattrisen

687. "95 Ways to find your first customers for customer development or your first sale" by Jason Evanish, *Building Customer Driven SaaS Products | Jason Evanish* – https://jasonevanish.com/2013/08/11/95-ways-to-find-your-first-customers-for-customer-development-or-your-first-sale/

688. *Superloops* – https://www.superloops.xyz/

689. *Twitter* – https://x.com/JohnPaulAndersn

690. *Twitter* – https://x.com/DevMolone

691. "Transistor - modern podcast hosting and analytics", *Transistor* – https://transistor.fm/

692. "BidTreat - Luxury Travel Deals from Charity Auctions", *Great Deals. Luxury Trips. Good Causes.* – https://www.bidtreat.com/

693. "Marie Strauchman (@mariestrauchman.bsky.social)", *Bluesky Social* – https://bsky.app/profile/mariestrauchman.bsky.social

694. "Nick Basile (@nick-basile.com)", *Bluesky Social* – https://bsky.app/profile/nick-basile.com

695. "Bluesky", *Bluesky Social* – https://bsky.app/profile/omarbytes.bsky.social

696. *Twitter* – https://x.com/gridpane

697. "Player feedback boards for game developers –", *Feature Upvote* – https://featureupvote.com/

698. https://twitter.com/Tony_Meijer

699. "Custom Cabinetry Made Simple", *Shop Cabinets* – https://www.hellobaru.com/

700. "Wim" by Wim Cools – https://thymer.com/wim

701. "Business Cartoon #6110", *Andertoons* – https://andertoons.com/business/cartoon/6110/i-dunno-lately-im-just-not-incentivized-

702. "Federal Holidays", *U.S. Office of Personnel Management* – https://www.opm.gov/policy-data-oversight/pay-leave/federal-holidays/

703. *37signals* – https://37signals.com/

704. "Fog Creek is now Glitch!" by Anil Dash, *Glitch: Make Better Software* – https://medium.com/make-better-software/fog-creek-is-now-glitch-8d0308aaf69e

705. https://www.pivotaltracker.com/

706. *Twitter* – https://x.com/ezsmith397/status/1760858039316332995?s=20

707. "Productivity Cartoon #7045", *Andertoons* – https://andertoons.com/productivity/cartoon/7045/moved-everything-from-to-do-list-to-it-is-what-it-is-list

708. "Prioritization is a Political Problem as Much as an Analytical Problem" by Richard Mironov, *Rich Mironov's Product Bytes* – https://www.mironov.com/pri-politics/

709. *Amazon* – https://www.amazon.com/Effortless-Experience-Conquering-Battleground-Customer/dp/1591845815

710. "The Blind Men and The Elephant: A Short Story about Perspective" by Kyle Kowalski, *Sloww* – https://www.sloww.co/blind-men-elephant/

711. "The New User Story Backlog is a Map", *We help you create successful product culture and process* – https://jpattonassociates.com/the-new-backlog/

712. "Opportunity Solution Tree", *ProductPlan* – https://www.productplan.com/glossary/opportunity-solution-tree/

713. "20 Product Prioritization Techniques: A Map and Guided Tour", *Folding Burritos* – https://foldingburritos.com/blog/product-prioritization-techniques/

714. "The Complete Guide to the Kano Model", *Folding Burritos* – https://foldingburritos.com/blog/kano-model/

715. "🥾 Reorganize your roadmap" by Adam Thornhill, *Podup* – https://podup.substack.com/p/reorganize-your-roadmap

716. "Chicken Cartoon #7546", *Andertoons* – https://andertoons.com/chicken/cartoon/7546/never-know-who-was-first-can-tell-you-whos-gonna-be-last

717. "Language Cartoon #7053", *Andertoons* – https://andertoons.com/language/cartoon/7053/thing-thought-was-thing-not-a-thing-but-other-thing-a-thing

718. "Laws of 10x found everywhere" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/10x/

719. "Stop chopping yourself to pieces" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/stop-chopping-to-pieces/

720. "Office Cartoon #3501", *Andertoons* – https://andertoons.com/office/cartoon/3501/i-am-multi-tasking-ive-done-this-report-three-or-four-times-already

721. "shreyasdoshi", *Linktree* – https://linktr.ee/shreyasdoshi

722. https://twitter.com/shreyas/status/1223816226918453253?s=20

723. https://twitter.com/paulg/status/1316661104911806465

724. "Underbelly: What haughty startup bloggers don't tell you" by Jason, *@ASmartBear* – https://blog.asmartbear.com/underbelly-what-haughty-startup-bloggers-dont-tell-you/

725. "What a startup does to you. Or: A celebration of new life" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/startup-life/

726. "Speed as a Habit" by First Round Staff, *First Round Review* – https://review.firstround.com/speed-as-a-habit/

727. "Jeff Bezos: There are 2 types of decisions to make, and don't confuse them" by Matt Rosoff, *Business Insider* – https://www.businessinsider.com/jeff-bezos-on-type-1-and-type-2-decisions-2016-4

728. "Sheep Cartoon #8790", *Andertoons* – https://andertoons.com/sheep/cartoon/8790/nothing-i-can-do-24-hours-police

729. "Vista Equity invests in Francisco's SmartBear, pushing valuation to $1.8bn-plus" by Milana Vinn, *PE Hub* – https://www.pehub.com/vista-equity-invests-in-franciscos-smartbear-pushing-valuation-to-1-8bn-plus/?utm_source=longform.asmartbear.com&utm_campaign=longform.asmartbear.com&utm_medium=post

730. "Zero-sum marketing channels: Good or bad for a startup to pursue?" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/zero-sum-marketing/

731. "How do I know where to advertise?" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/begin-advertising/

732. "Distinguishing constructive criticism from bad business advice" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/bad-advice/

733. "If Kindergarten were like Social Media Marketing" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/social-media-marketing/

734. "Advice Cartoon #7603", *Andertoons* – https://andertoons.com/advice/cartoon/7603/we-used-to-sit-on-peoples-shoulders-give-conflicting-advice?searchid=44934

735. "Easy to criticize, hard to create" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/criticize-create/

736. "Real Estate Cartoon #8416", *Andertoons* – https://andertoons.com/realestate/cartoon/8416/location-separates-from-competition

737. "Prisoner's dilemma", *Wikipedia* – https://en.wikipedia.org/wiki/Prisoner%27s_dilemma

738. "Infographic: Attack of the Clones" by Felix Richter, *Statista Daily Data* – https://www.statista.com/chart/10558/daus-instagram-stories-whatsapp-status-snapchat/?__sso_cookie_checker=failed

739. *Amazon* – https://www.amazon.com/7-Powers-Foundations-Business-Strategy/dp/0998116319

740. "A Taxonomy of Moats" by Kirk Willard, *Reaction Wheel* – https://reactionwheel.net/2019/09/a-taxonomy-of-moats.html

741. "Practical estimation of cloud storage costs for clinical genomic data" by Niklas Krumm, Noah Hoffman, *ScienceDirect* – https://www.sciencedirect.com/science/article/pii/S2352551719301052

742. "AWS growth accelerates in quarter marred by outages" by Jordan Novet, *CNBC* – https://www.cnbc.com/2022/02/03/amazon-web-services-earnings-q4-2021.html

743. "Amazon Web Services posts record $13.5B in *profits* for 2020 in Andy Jassy's AWS swan song" by Todd Bishop, *GeekWire* – https://www.geekwire.com/2021/amazon-web-services-posts-record-13-5b-profits-2020-andy-jassys-aws-swan-song/

744. https://www.theinformation.com/articles/why-aws-makes-money-and-google-cloud-doesnt

745. "Decoder with Nilay Patel: Recode Decode: Amazon Web Services CEO Andy Jassy (Live at Code 2019) (51m)", *Castro 2* – https://castro.fm/episode/uQph8q

746. "Fish Cartoon #5903", *Andertoons* – https://andertoons.com/fish/cartoon/5903/i-told-you-to-order-piranhas-earlier

747. "Weight Cartoon #7574", *Andertoons* – https://andertoons.com/weight/cartoon/7574/im-not-letting-myself-go-im-scaling

748. "File:Cynefin framework 2022.jpg - Wikimedia Commons", *Wikipedia* – https://commons.wikimedia.org/wiki/File:Cynefin_framework_2022.jpg

749. "Cynefin framework", *Wikipedia* – https://en.wikipedia.org/wiki/Cynefin_framework

750. "Startups are failing because they make products no one wants" by Erin Griffith, *Fortune* – https://fortune.com/2014/09/25/why-startups-fail-according-to-their-founders/

751. "How Not to Die" – https://www.paulgraham.com/die.html

752. "Why Startups Fail, According to Their Founders" by Erin Griffith, *Entrepreneur* – https://www.entrepreneur.com/business-news/why-startups-fail-according-to-their-founders/238088

753. "Pirate Cartoon #5924", *Andertoons* – https://andertoons.com/pirate/cartoon/5924/i-dunno-first-mate-has-been-done-to-death-how-about-this-co-pirate

754. "News Cartoon #4713", *Andertoons* – https://andertoons.com/news/cartoon/4713/surprised-investors-expecting-worst-performing-miserably

755. "More money if you do, more money if you don't" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/more-money/

756. "Weather Cartoon #5221", *Andertoons* – https://andertoons.com/weather/cartoon/5221/but-to-be-fair-theres-fifty-percent-chance-of-just-about-anything

757. "Insect Cartoon #4408", *Andertoons* – https://andertoons.com/insect/cartoon/4408/so-where-do-you-see-yourself-in-ten-minutes

758. "Marketing at early startups: Deep, not wide" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/one-channel/

759. "Business Cartoon #5966", *Andertoons* – https://andertoons.com/business/cartoon/5966/i-dont-get-it-weve-got-mission-statement-credo-and-mantra

760. "Why Slack Hurts Your Teams Productivity" by Ben Bartling, *Medium* – https://benbartling.medium.com/why-slack-hurts-your-teams-productivity-af9804d06606

761. "Reddit Inc Blog Homepage", *Upvoted* – https://redditinc.com/blog

762. "Nick Craver", *HTTPS on Stack Overflow: The End of a Long Road* – https://nickcraver.com/blog/2017/05/22/https-on-stack-overflow/

763. "How WIRED Completely Encrypted Itself" by Zack Tollman, *WIRED* – https://www.wired.com/2016/09/wired-completely-encrypted/

764. "Hickam's dictum", *Wikipedia* – https://en.wikipedia.org/wiki/Hickam%27s_dictum

765. "Five whys", *Wikipedia* – https://en.wikipedia.org/wiki/Five_whys

766. "Startups Don't Die, They Commit Suicide" by Contributor, *TechCrunch* – https://techcrunch.com/2011/06/27/startups-dont-die-they-commit-suicide/

767. "Science Cartoon #7252", *Andertoons* – https://andertoons.com/science/cartoon/7252/correlation-does-not-imply-causation-we-know-it-was-brian

768. "Holacracy", *Wikipedia* – https://en.wikipedia.org/wiki/Holacracy

769. "Teal organisation", *Wikipedia* – https://en.wikipedia.org/wiki/Teal_organisation

770. "Gadget Cartoon #6598", *Andertoons* – https://andertoons.com/gadget/cartoon/6598/room-for-new-device-between-mini-tablet-and-mega-smartphone

771. "Continuous Innovation Platform" – https://leanstack.com/

772. "About LEANSTACK" – https://leanstack.com/about

773. "Subject-matter expert", *Wikipedia* – https://en.wikipedia.org/wiki/Subject-matter_expert

774. "Success Cartoon #7852", *Andertoons* – https://andertoons.com/success/cartoon/7852/wonder-what-mother-a-simple-dairy-cow-would-have-made-this

775. "Hell Cartoon #8967", *Andertoons* – https://andertoons.com/hell/cartoon/8967/no-fan-heat-toil-but-paperwork-gets-me

776. "Bear", *Bear Markdown Notes* – https://bear.app/

777. "Personal data", *Wikipedia* – https://en.wikipedia.org/wiki/Personal_data

778. "Pareto efficiency", *Wikipedia* – https://en.wikipedia.org/wiki/Pareto_efficiency

779. "Threads" – https://www.threads.com/@mariyadelano/post/C6JqmIEL5P8

780. "Business Cartoon #4018", *Andertoons* – https://andertoons.com/business/cartoon/4018/im-sorry-jim-im-afraid-this-only-win-win-situation

781. "Office Cartoon #3501", *Andertoons* – https://andertoons.com/office/cartoon/3501/i-am-multi-tasking-ive-done-this-report-three-or-four-times-already

782. "Theory of Constraints 101: Table of Contents" by Tiago Forte, *Forte Labs* – https://fortelabs.com/blog/theory-of-constraints-101-table-of-contents/

783. "Theory of Constraints 107: The Five Focusing Steps" by Tiago Forte, *Forte Labs* – https://fortelabs.com/blog/theory-of-constraints-107-identifying-the-constraint/

784. "Theory of Constraints 106: DBR at Microsoft" by Tiago Forte, *Forte Labs* – https://fortelabs.com/blog/theory-of-constraints-106-dbr-at-microsoft/

785. "Theory of Constraints 109: Optimizing the Constraint" by Tiago Forte, *Forte Labs* – https://fortelabs.com/blog/theory-of-constraints-109-the-psychology-of-subordination/

786. "Theory of Constraints 102: The Illusion of Local Optima" by Tiago Forte, *Forte Labs* – https://fortelabs.com/blog/theory-of-constraints-102-local-optima/

787. "Helping your customers navigate change" by Beau Ulrey, *UX Collective* – https://uxdesign.cc/help-your-customer-navigate-change-1bb674f6ce23?gi=b4127f9f7f30

788. "Technology Cartoon #6637", *Andertoons* – https://andertoons.com/technology/cartoon/6637/culmination-suggestions-employees-focus-consultants-its-terrible

789. https://twitter.com/shreyas

790. https://twitter.com/shreyas/status/1249039638829793280

791. 0, *Lenny Rachitsky* – https://www.lennyrachitsky.com/

792. "The Twenty Minute VC (20VC): 20 Product: Lenny Rachitsky on The 3 Key Roles of the Product Manager, 5 Skills All The Best PMs Have, When To Hire Your First PM, How to Structure the Hiring Process for PMs & What Leaders Can Do to Make Their PMs Successful (42m)", *Castro 2* – https://castro.fm/episode/UzVQdL

793. https://cdn.substack.com/image/fetch/f_auto,q_auto:good,fl_progressive:steep/https%3A%2F%2Fbucketeer-e05bbc84-baa3-437e-9518-adb32be77984.s3.amazonaws.com%2Fpublic%2Fimages%2F72e7fff6-10dc-429e-bd57-2a2a9d8d9711_2048x1654.png

794. "Productside" by Zelko Marinovic, *Product Management Courses | Productside* – https://productside.com/

795. "Caveman Cartoon #8678", *Andertoons* – https://andertoons.com/caveman/cartoon/8678/fire-good-except-sometimes-bucket-of-water

796. "If You Don't Think You Need It, You Haven't Seen Greatness" by Contributor, *TechCrunch* – https://techcrunch.com/2012/07/24/you-havent-seen-greatness/

797. "The Office of Strategy Management" by Robert S. Kaplan, *Harvard Business Review* – https://hbr.org/2005/10/the-office-of-strategy-management

798. "Easter Cartoon #6570", *Andertoons* – https://andertoons.com/easter/cartoon/6570/i-didnt-say-i-was-unfulfilled-i-said-i-was-hollow

799. Derek Lin, *Taoism.net* – https://taoism.net/

800. "What a startup does to you. Or: A celebration of new life" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/startup-life/

801. "News Cartoon #4033", *Andertoons* – https://andertoons.com/news/cartoon/4033/reaction-to-news-was-mixed-largely-because-we-asked-more-than-one-person

802. "Deep dive: Cancellation rate in SaaS business models" by Jason, *@ASmartBear* – https://blog.asmartbear.com/cancellation-rate-in-saas-business-models/

803. "Laws of 10x found everywhere" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/10x/

804. "The rise of the "successful" unsustainable company" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/unsustainable-companies/

805. "The unprofitable SaaS business model trap" by Jason, *@ASmartBear* – https://blog.asmartbear.com/unprofitable-saas-business-model/

806. "A Quick Primer on the Rule of 40" by Sam Baker, *Scale Venture Partners* – https://www.scalevp.com/insights/a-quick-primer-on-the-rule-of-40/

807. "Politics Cartoon #1515", *Andertoons* – https://andertoons.com/politics/cartoon/1515/no-its-too-late-for-oversight-committee-send-it-to-hindsight-committee

808. "Lindy effect", *Wikipedia* – https://en.wikipedia.org/wiki/Lindy_Effect

809. *Twitter* – https://x.com/dmytrokrasun/status/1800196093873164396

810. "Form S-1" – https://www.sec.gov/Archives/edgar/data/1609711/000119312514230425/d728713ds1.htm

811. "GoDaddy Revenue 2013-2025", *MacroTrends* – https://www.macrotrends.net/stocks/charts/GDDY/godaddy/revenue

812. "Technology Cartoon #6125", *Andertoons* – https://andertoons.com/technology/cartoon/6125/i-dunno-kind-of-defeats-purpose-doesnt-it

813. "Easy to criticize, hard to create" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/criticize-create/

814. "The Feynman Lectures on Physics", *Wikipedia* – https://en.wikipedia.org/wiki/The_Feynman_Lectures_on_Physics

815. "The mid-market briar patch" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/mid-market/

816. https://twitter.com/Rainmaker1973/status/1638802766628880385

817. https://faculty.washington.edu/moishe/branko/BG228.SymmetricVenn.diagrams.pdf

818. *Twitter* – https://x.com/phuctm97/status/1745330740576043354?s=20

819. "WP Engine, the Smarter Way to Host WordPress", *WP Engine* – https://wpengine.com/

820. "Better for whom?" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/better-for-whom/

821. https://itwatchdogs.com/

822. "Home", *Webstock* – https://www.webstock.org.nz/

823. https://www.hipmunk.com/

824. "Affordances and Metaphors" by Joel Spolsky, *Joel on Software* – https://www.joelonsoftware.com/2000/04/18/affordances-and-metaphors/

825. Home, *Alexis Ohanian Sr.* – https://alexisohanian.com/

826. "Agile Marketing: Pucha Kecha format", *YouTube* – https://www.youtube.com/watch?si=EuOMtmhvKEQVlMGN&v=YuXmM8bHMxY&feature=youtu.be

827. "Every Time Zone Converter", *Every Time Zone* – https://everytimezone.com/

828. "Build Your Business, One Win At A Time", *Stacking the Bricks* – https://stackingthebricks.com/?_r=uf

829. "Writing Decisions: Headline tests on the Highrise signup page" by Adrian Bye, *Signal v. Noise by Basecamp* – https://signalvnoise.com/posts/1525-writing-decisions-headline-tests-on-the-highrise-signup-page

830. "Shark Cartoon #5310", *Andertoons* – https://andertoons.com/shark/cartoon/5310/as-long-as-it-has-all-its-fins-and-remorseless-eating-machine-well-be-happy

831. https://longform.asmartbear.com/ref/life-after-exit-en.pdf

832. "News Cartoon #4713", *Andertoons* – https://andertoons.com/news/cartoon/4713/surprised-investors-expecting-worst-performing-miserably

833. "Law Firm Succession/Exit Strategies: Valuation of the Firm", *Olmstead and Associates* – https://www.olmsteadassoc.com/resource-center/law-firm-succession-exit-strategies-valuing-the-firm/

834. "YouTube" – https://www.youtube.com/embed/O4MtQGRIIuA?start=267

835. "Change Cartoon #7899", *Andertoons* – https://andertoons.com/change/cartoon/7899/second-quarter-is-up-no-fourth-quarter-now-second-again

836. https://dilbertblog.typepad.com/the_dilbert_blog/2007/07/career-advice.html

837. *Twitter* – https://x.com/TaylorPearsonMe

838. "How to Get Focused", *Taylor Pearson* – https://taylorpearson.me/how-to-get-focused/

839. "Three Focus Failures and How to Avoid Them", *Taylor Pearson* – https://taylorpearson.me/three-focus-failures/

840. "VC Fund Returns Are More Skewed Than You Think", *VC Adventure* – https://sethlevine.com/archives/2020/10/vc-fund-returns-are-more-skewed-than-you-think.html

841. "Why 95% of Traders Lose Money (And That's in a Bull Market)" by Sean McCloskey, *Energy & Capital* – https://www.energyandcapital.com/why-95-of-traders-lose-money-and-thats-in-a-bull-market/

842. "Police Cartoon #6040", *Andertoons* – https://andertoons.com/police/cartoon/6040/bad-news-something-that-happened-in-vegas-has-been-spotted-in-des-moines

843. *Amazon* – https://www.amazon.com/gp/product/1430210788?ie=UTF8&amp%3Btag=asmbe-20&amp%3BlinkCode=as2&amp%3Bcamp=1789&amp%3Bcreative=390957&amp%3BcreativeASIN=1430210788

844. "Archive Interviews", *Business Podcast for Startups* – https://mixergy.com/interviews/

845. "Why startup biz dev deals almost never get done" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/startup-biz-dev/

846. "Shark Cartoon #5310", *Andertoons* – https://andertoons.com/shark/cartoon/5310/as-long-as-it-has-all-its-fins-and-remorseless-eating-machine-well-be-happy

847. "Do 'maximisers' or 'satisficers' make better decisions?" by Bryan Lufkin, *BBC* – https://www.bbc.com/worklife/article/20210329-do-maximisers-or-satisficers-make-better-decisions

848. "Game Cartoon #8851", *Andertoons* – https://andertoons.com/game/cartoon/8851/chess-bishop-qbert-dating-moves

849. "Business Cartoon #3719", *Andertoons* – https://andertoons.com/business/cartoon/3719/no-its-fine-ive-just-never-seen-coupons-used-in-an-acquisition

850. "Crime Cartoon #9091", *Andertoons* – https://andertoons.com/crime/cartoon/9091/stolen-identity-living-best-life

851. "Memory Cartoon #7810", *Andertoons* – https://andertoons.com/memory/cartoon/7810/were-going-to-look-back-on-false-memories-of-this-and-laugh

852. "Egg Cartoon #6898", *Andertoons* – https://andertoons.com/egg/cartoon/6898/not-interested-putting-together-concerned-about-giant-chicken

853. "Marketing Platform Independence" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/marketing-platform-independence/

854. "Marketing at early startups: Deep, not wide" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/one-channel/

855. "How to find that first big customer" by Jason, *@ASmartBear* – https://blog.asmartbear.com/find-first-big-customer/

856. "Imbalanced People" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/imbalanced-people/

857. https://en.wikipedia.org/wiki/Wetware_(brain

858. "Love Cartoon #6341", *Andertoons* – https://andertoons.com/love/cartoon/6341/what-happens-to-the-supply-chain-if-you-and-debbie-break-up

859. "Science Cartoon #8676", *Andertoons* – https://andertoons.com/science/cartoon/8676/fine-stand-there-silent-have-the-lab-until-9

860. "FDA panel says common over-the-counter decongestant doesn't work" by Berkeley Lovelace Jr., *NBC News* – https://www.nbcnews.com/health/health-news/fda-panel-says-common-counter-decongestant-phneylephrine-doesnt-work-rcna104424

861. "Why Has a Useless Cold Medication Been Allowed on Shelves for Years?" by Sarah Zhang, *The Atlantic* – https://www.theatlantic.com/health/archive/2023/09/cold-medicine-decongestant-phenylephrine-ineffective/675303/

862. "6 Drugs Approved Despite Failed Trials or Minimal Data" by Heather McKenzie, *BioSpace* – https://www.biospace.com/6-drugs-approved-despite-failed-trials-or-minimal-data

863. "Replication crisis", *Wikipedia* – https://en.wikipedia.org/wiki/Replication_crisis

864. "Nobel laureate challenges psychologists to clean up their act" by Ed Yong, *Nature* – https://www.nature.com/articles/nature.2012.11535?error=cookies_not_supported&code=c3bc81f0-5c49-4b17-87c5-b166e6880ac6

865. "Mystery of a broken A/B test", *YouTube* – https://www.youtube.com/watch?v=FaUO2-AQmr0&feature=youtu.be

866. "p-Hacking in A/B Testing" by See all articles by Ron Berman – https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3204791

867. "When a paper becomes a sensation." by Ron, *Ron Berman @ Wharton* – https://www.ron-berman.com/2018/12/17/when-a-paper-becomes-a-sensation/

868. "Archie Abrams" by Lenny's Newsletter, *Lenny's Newsletter* – https://www.lennysnewsletter.com/p/shopifys-growth-archie-abrams

869. "Threads" – https://www.threads.com/@dilemmaworker/post/DDNl-1nMD3y

870. "Result Cartoon #6798", *Andertoons* – https://andertoons.com/result/cartoon/6798/you-wont-believe-it-experiment-the-results-theyre-conclusive

871. *Twitter* – https://x.com/einarvollset

872. "Lazy Cartoon #3123", *Andertoons* – https://andertoons.com/lazy/cartoon/3123/could-you-at-least-burn-candle-at-one-end

873. "Shoes, Sneakers, Boots, & Clothing + FREE SHIPPING" by Free People, *Zappos.com* – https://www.zappos.com/

874. https://twitter.com/pronounced_kyle/status/1768852493092680036

875. "TypeRacer - the global typing competition", *TypeRacer* – https://play.typeracer.com/

876. https://www.43folders.com/2005/02/18/quick-tips-on-processing-your-email-inbox

877. "RescueTime" – https://www.rescuetime.com/

878. https://www.43folders.com/izero

879. "Work Cartoon #6885", *Andertoons* – https://andertoons.com/work/cartoon/6885/boost-productivity-by-making-everyone-work-harder

880. "Cow Cartoon #8608", *Andertoons* – https://andertoons.com/cow/cartoon/8608/another-unanimous-vote-herd-mentality

881. https://www.michaelmauboussin.com/michael_mauboussin.htm

882. "Jack L. Treynor", *Wikipedia* – https://en.wikipedia.org/wiki/Jack_L._Treynor

883. "Addition Cartoon #8376", *Andertoons* – https://andertoons.com/addition/cartoon/8376/math-problem-error-convince-grade-take-seriously

884. "Teacher Cartoon #6309", *Andertoons* – https://andertoons.com/teacher/cartoon/6309/its-not-cheating-its-crowdsourcing

885. "How to be a millionaire", *Seth's Blog* – https://seths.blog/2006/12/how_to_be_a_mil/

886. https://twitter.com/JamesSurowiecki

887. "The Wisdom of Crowds", *Wikipedia* – https://en.wikipedia.org/wiki/The_Wisdom_of_Crowds

888. https://press.princeton.edu/books/paperback/9780691138541/the-difference?srsltid=AfmBOort5io_jYZf051JgrXTdztrPyA8JKAyD9asebniZzR-yx1_YW2-

889. "Valentine Cartoon #9174", *Andertoons* – https://andertoons.com/valentine/cartoon/9174/hit-her-with-lower-expectations-first

890. "Why business blogs should focus on cheerleaders, not lead-generation" by Jason, *@ASmartBear* – https://blog.asmartbear.com/why-business-blogs-should-focus-on-cheerleaders-not-lead-generation/

891. "You're a little company, now act like one" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/little-company/

892. "Thales of Miletus", *Wikipedia* – https://en.wikipedia.org/wiki/Thales

893. "Don't write a business plan" by Jason, *@ASmartBear* – https://blog.asmartbear.com/business-plan/

894. "Meetings Cartoon #7019", *Andertoons* – https://andertoons.com/meetings/cartoon/7019/youve-got-meetings-and-then-pencilled-in-self-immolation

895. "Dog Cartoon #6812", *Andertoons* – https://andertoons.com/dog/cartoon/6812/i-was-sure-something-over-there-needed-to-bark-at

896. "Book Cartoon #5328", *Andertoons* – https://andertoons.com/book/cartoon/5328/go-back-and-give-that-part-little-more-oomph-really-make-me-believe

897. *Amazon* – https://www.amazon.com/Creative-Conspiracy-Rules-Breakthrough-Collaboration/dp/1422173348?tag=wwwfccom-20

898. "Virtual communication curbs creative idea generation" by Melanie S. Brucks, *Nature* – https://www.nature.com/articles/s41586-022-04643-y?error=cookies_not_supported&code=05ff63b6-7914-4ff4-9536-debf15ddf826

899. "WP Engine, the Smarter Way to Host WordPress", *WP Engine* – https://wpengine.com/

900. "Pick one and own it" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/one-benefit/

901. "The Complete Guide to the Kano Model", *Folding Burritos* – https://foldingburritos.com/blog/kano-model/

902. "Computerworld", *Google Books* – https://books.google.com/books?id=jNkvgvGj198C&pg=PA2&lpg=PA2&dq=meaning+of+%22freeze+the+market%22+press+release&source=bl&ots=SZXMvh35zf&sig=ACfU3U1H306Av97xzkykJFEKXgOzi9Gogg&hl=en&sa=X&ved=2ahUKEwjX_JGf4fv3AhVsBDQIHYWoAN8Q6AF6BAgfEAM

903. "Competition Cartoon #7904", *Andertoons* – https://andertoons.com/competition/cartoon/7904/vendors-one-sells-balloons-the-other-sharp-little-pins

904. "Lion Cartoon #8997", *Andertoons* – https://andertoons.com/lion/cartoon/8997/what-do-you-have-sick-injured

905. "Darwinian advice for "Going Viral"" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/going-viral/

906. "The Federal Reserve's Little Secret" by Rogé Karma, *The Atlantic* – https://www.theatlantic.com/ideas/archive/2024/06/interest-rates-inflation/678802/

907. "Loss aversion", *Wikipedia* – https://en.wikipedia.org/wiki/Loss_aversion

908. "Endowment effect", *Wikipedia* – https://en.wikipedia.org/wiki/Endowment_effect

909. https://academic.oup.com/jcr/article-abstract/27/3/360/1796841

910. "Sunk cost", *Wikipedia* – https://en.wikipedia.org/wiki/Sunk_cost
911. "Math Cartoon #3618", *Andertoons* – https://andertoons.com/math/cartoon/3618/if-two-negatives-make-positive-how-come-two-wrongs-dont-make-right
912. "Availability heuristic", *Wikipedia* – https://en.wikipedia.org/wiki/Availability_heuristic
913. "Confirmation bias", *Wikipedia* – https://en.wikipedia.org/wiki/Confirmation_bias
914. "Overconfidence effect", *Wikipedia* – https://en.wikipedia.org/wiki/Overconfidence_effect
915. "Are we all less risky and more skillful than our fellow drivers?" by Ola Svenson *, *ScienceDirect* – https://www.sciencedirect.com/science/article/abs/pii/0001691881900056
916. "Easy to criticize, hard to create" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/criticize-create/
917. "Serial-position effect", *Wikipedia* – https://en.wikipedia.org/wiki/Serial_position_effect
918. "Survivorship bias", *Wikipedia* – https://en.wikipedia.org/wiki/Survivorship_bias
919. "Herd behavior", *Wikipedia* – https://en.wikipedia.org/wiki/Herd_behavior
920. "Status quo bias", *Wikipedia* – https://en.wikipedia.org/wiki/Status_quo_bias
921. "Animal Cartoon #4904", *Andertoons* – https://andertoons.com/animal/cartoon/4904/its-gazelles-they-got-restraining-order
922. "Sex Cartoon #5912", *Andertoons* – https://andertoons.com/sex/cartoon/5912/my-wife-my-duplicate
923. "Eudora Welty", *Wikipedia* – https://en.wikipedia.org/wiki/Eudora_Welty
924. "Eudora Welty: Why I Live at the P.O." – https://art-bin.com/art/or_weltypostoff.html
925. "Copy and paste styles (paintbrush tools)", *company logo* – https://help.marq.com/paintbrush-tool
926. "Hello, I'm 1074018628" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/hello-support/
927. "Seth's Blog", *Seth's Blog* – https://seths.blog/
928. *Amazon* – https://www.amazon.com/Not-That-You-Asked-Obsessions/dp/0812977599/ref=sr_1_2?ie=UTF8&s=books&qid=1234733932&sr=1-2
929. *James Altucher* – https://www.jamesaltucher.com/
930. https://www.jamesaltucher.com/2011/08/girls-girls-girls/
931. https://bit.ly/qWDNuZ
932. "Book Cartoon #4820", *Andertoons* – https://andertoons.com/book/cartoon/4820/ventriloquist-dummy-reading-book-entitled-ventriloquism-for-dummies
933. "Imbalanced People" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/imbalanced-people/
934. "Superheroes Cartoon #7776", *Andertoons* – https://andertoons.com/superheroes/cartoon/7776/today-were-going-to-start-with-league-building-exercises
935. "Author Cartoon #9207" by Author Cartoon #9207, *Andertoons* – https://andertoons.com/author/cartoon/9207/jargon-buzzword-lingo
936. "If You Say Something Is "Likely," How Likely Do People Think It Is?" by Andrew Mauboussin, *Harvard Business Review* – https://hbr.org/2018/07/if-you-say-something-is-likely-how-likely-do-people-think-it-is
937. "datasets/Perception-of-Probability-Words at master · wadefagen/datasets" by wadefagen, *GitHub* – https://github.com/wadefagen/datasets/tree/master/Perception-of-Probability-Words
938. "Perception of Probability Words" by Wade Fagen-Ulmschneider – https://waf.cs.illinois.edu/visualizations/Perception-of-Probability-Words/
939. "Probability Cartoon #7935", *Andertoons* – https://andertoons.com/probability/cartoon/7935/a-is-more-likely-but-i-gotta-say-feel-like-b-wants-it-more
940. "Business Cartoon #3719", *Andertoons* – https://andertoons.com/business/cartoon/3719/no-its-fine-ive-just-never-seen-coupons-used-in-an-acquisition

941. "It's official: Microsoft is buying Minecraft developer Mojang for $2.5 billion" by Ben Woods, *The Next Web* – https://thenextweb.com/news/official-microsoft-buying-minecraft-developer-mojang

942. "'I felt lost and empty': Founders share their exit blues stories" by Kai Nicol-Schwarz, *Sifted* – https://sifted.eu/articles/founders-exit-blues

943. "Dealing with the Emotional Fallout of Selling Your Business" by Jeff Giesea, *Harvard Business Review* – https://hbr.org/2015/09/dealing-with-the-emotional-fallout-of-selling-your-business

944. "I am rich and have no idea what to do with my life" – https://vinay.sh/i-am-rich-and-have-no-idea-what-to-do-with-my-life/

945. *Twitter* – https://x.com/notch/status/637562496056995840

946. "Mental health in elite athletes: International Olympic Committee consensus statement (2019)", *British Journal of Sports Medicine* – https://bjsm.bmj.com/content/53/11/667

947. "IOC Mental Health in Elite Athletes Toolkit" – https://www.olympics.com/athlete365/articles/mentally-fit/ioc-mental-health-in-elite-athletes-toolkit-x3908

948. "Holly Bradshaw on being an IOC Mental Heath Ambassador and the post-Olympic blues", *Team GB* – https://www.teamgb.com/article/ioc-mental-health-ambassador-holly-bradshaw-opens-up-about-post-olympic/6gBc8IKg5GSMNydsQW3IuD

949. "What a startup does to you. Or: A celebration of new life" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/startup-life/

950. https://onlinelibrary.wiley.com/doi/full/10.1002/hbm.23562

951. https://longform.asmartbear.com/ref/life-after-exit-en.pdf

952. https://wpengine.com/2012/11/creating-a-company-culture-that-thrills-customers-at-wp-engine/

953. "Meeting Cartoon #5327", *Andertoons* – https://andertoons.com/meeting/cartoon/5327/we-all-agree-lets-go-back-to-desks-discuss-why-this-wont-work

954. https://twitter.com/asmartbear/status/1529105502679998465

955. "Meeting Cartoon #6071", *Andertoons* – https://andertoons.com/meeting/cartoon/6071/it-seems-ive-run-out-of-sports-analogies-so-meeting-adjourned

956. "Dance Cartoon #2108", *Andertoons* – https://andertoons.com/dance/cartoon/2108/no-ones-winning-its-ballet

957. "Business Cartoon #4018", *Andertoons* – https://andertoons.com/business/cartoon/4018/im-sorry-jim-im-afraid-this-only-win-win-situation

958. "Home", *Chief Marketing Technologist* – https://chiefmartec.com/

959. "Marketing Cartoon #7220", *Andertoons* – https://andertoons.com/marketing/cartoon/7220/i-miss-the-old-days-when-we-could-just-tout-things

960. "Marketing Platform Independence" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/marketing-platform-independence/

961. "Marketing at early startups: Deep, not wide" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/one-channel/

962. "Graph Cartoon #7231", *Andertoons* – https://andertoons.com/graph/cartoon/7231/serendipity-is-up-fluke-doing-well-concerned-about-dumb-luck

963. "Ecclesiastes 9:11 - Enjoy Your Portion in This Life", *Bible Hub* – https://biblehub.com/ecclesiastes/9-11.htm

964. *Twitter* – https://x.com/_baretto/status/1856290846850854974

965. "Psychiatry Cartoon #6059", *Andertoons* – https://andertoons.com/psychiatry/cartoon/6059/what-luck-one-of-my-personalities-psychiatrist

966. "Strong opinions, somewhat weakly held" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/strong-opinions-weakly-held/

967. "Schedule Cartoon #7498", *Andertoons* – https://andertoons.com/schedule/cartoon/7498/youve-got-boring-at-11-00-pointless-at-1-00-and-interminable-at-3-00?searchid=44787

968. "Gypsy Cartoon #7292", *Andertoons* — https://andertoons.com/gypsy/cartoon/7292/nothing-interesting-in-any-previous-lives-either-consistent?searchid=44788
969. "Sales Cartoon #8669", *Andertoons* — https://andertoons.com/sales/cartoon/8669/lesson-learned-no-one-listen-to-don
970. "Dog Cartoon #5038", *Andertoons* — https://andertoons.com/dog/cartoon/5038/whats-that-boy-paradigm-shift
971. ""Premature optimization is the root of all evil" More and more, I'm realizing th...", *Hacker News* — https://news.ycombinator.com/item?id=23096321
972. https://twitter.com/hveldstra/status/1750900705634742423
973. "Insect Cartoon #8816", *Andertoons* — https://andertoons.com/insect/cartoon/8816/chrysalis-late-butterfly
974. *Amazon* — https://www.amazon.com/Windows-95-Bug-Collection-Work-Arounds/dp/0201489953
975. "Family Cartoon #311", *Andertoons* — https://andertoons.com/family/cartoon/311/as-you-can-see-weve-not-only-kept-up-with-jones-but-surpassed-them-next-up-nelsons
976. https://twitter.com/hveldstra
977. "Security Cartoon #1505", *Andertoons* — https://andertoons.com/security/cartoon/1505/two-desks-one-reads-security-desk-other-reads-insecurity-desk
978. "Matt the Electrician" — https://matttheelectrician.com/
979. "WP Engine, the Smarter Way to Host WordPress", *WP Engine* — https://wpengine.com/
980. "WP Engine, the Smarter Way to Host WordPress", *WP Engine* — https://wpengine.com//
981. *Rob Walling* — https://robwalling.com/
982. "Kalzumeus Software", *Kalzumeus* — https://www.kalzumeus.com/
983. "Ghost Cartoon #7196", *Andertoons* — https://andertoons.com/ghost/cartoon/7196/oh-thats-the-looming-specter-of-dwindling-profits-ignore-him
984. "Sales Cartoon #2015", *Andertoons* — https://andertoons.com/sales/cartoon/2015/couldnt-we-just-move-bottom-line-down-little
985. "Goldilocks Cartoon #8067", *Andertoons* — https://andertoons.com/goldilocks/cartoon/8067/case-two-kids-chow-down-on-witchs-house-then-burn-her-alive
986. "Back to the Story Spine" by Aerogramme Writers' Studio, *Aerogramme Writers' Studio* — https://www.aerogrammestudio.com/2013/06/05/back-to-the-story-spine/
987. https://twitter.com/theandreboso/status/1713910012051865602
988. "Double Diamond (design process model)", *Wikipedia* — https://en.wikipedia.org/wiki/Double_Diamond_%28design_process_model%29
989. "Sales Cartoon # 8412", *Andertoons* — https://andertoons.com/sales/cartoon/8412/understand-burden-exposition-just-see-numbers
990. "Goal Cartoon #8072", *Andertoons* — https://andertoons.com/goal/cartoon/8072/i-think-i-see-why-were-not-getting-anywhere
991. "Story Cartoon #8705", *Andertoons* — https://andertoons.com/story/cartoon/8705/witch-old-lady-children-deal
992. *Amazon* — https://www.amazon.com/Goal-Process-Ongoing-Improvement/dp/0884271951
993. "Teamwork Cartoon #8070", *Andertoons* — https://andertoons.com/teamwork/cartoon/8070/ok-but-if-we-work-together-whammodepth-perception
994. https://twitter.com/JamesSurowiecki
995. "Puzzles Cartoon #7887", *Andertoons* — https://andertoons.com/puzzles/cartoon/7887/weve-got-three-sides-now-i-think-were-making-good-progress
996. "Book Cartoon #8410", *Andertoons* — https://andertoons.com/book/cartoon/8410/once-upon-time-lazy-writing-go-ahead
997. "Sex Cartoon #1202", *Andertoons* — https://andertoons.com/sex/cartoon/1202/i-understand-that-part-but-then-where-do-storks-come-from
998. "Rackspace Technology", *Wikipedia* — https://en.wikipedia.org/wiki/Rackspace_Technology

999. "Earnings before interest, taxes, depreciation and amortization", *Wikipedia* – https://en.wikipedia.org/wiki/Earnings_before_interest,_taxes,_depreciation_and_amortization

1000. "SaaS Metrics 2.0 - A Guide to Measuring and Improving what Matters" by David Skok, *For Entrepreneurs* – https://www.forentrepreneurs.com/saas-metrics-2/

1001. "COLOR 101" by Nekiah Torres, *Blogger* – https://fromthereztothecity.blogspot.com/2013/07/color-101.html?m=1

1002. "Color wheel", *Wikipedia* – https://en.wikipedia.org/wiki/Color_wheel

1003. "Electromagnetic Spectrum", *Mini Physics* – https://www.miniphysics.com/electromagnetic-spectrum.html

1004. "Johann Wolfgang von Goethe", *Wikipedia* – https://en.wikipedia.org/wiki/Johann_Wolfgang_von_Goethe

1005. "Cone cell", *Wikipedia* – https://en.wikipedia.org/wiki/Cone_cell

1006. https://luminusdevices.zendesk.com/hc/en-us/articles/4414846186253-What-is-the-CIE-Color-Space-What-s-the-difference-between-CIE-1931-and-CIE-1976

1007. "Color Space Confusion" by Jeff Yurek, *dot color* – https://dot-color.com/2012/08/14/color-space-confusion/

1008. "Josef Albers exhibition explores how colors are perceived", *www.sbc.edu* – https://www.sbc.edu/live/news/1777-josef-albers-exhibition-explores-how-colors-are

1009. "Color constancy", *Wikipedia* – https://en.wikipedia.org/wiki/Color_constancy

1010. "handprint : light and the eye" – https://www.handprint.com/HP/WCL/color1.html

1011. "Inside the Publishing Revolution: The Adobe Story" by Pamela Pfiffner, *Peachpit* – https://www.peachpit.com/store/inside-the-publishing-revolution-the-adobe-story-9780321115645

1012. "Best Kept Secrets of Code Review", *Login* – https://smartbear.com/resources/ebooks/best-kept-secrets-of-code-review/

1013. "Sheep Cartoon #8415", *Andertoons* – https://andertoons.com/sheep/cartoon/8415/sheep-tennis-ball-change-status-quo

1014. "Profit Whale Curve", *YouTube* – https://www.youtube.com/watch?v=g9-TVWNu4Aw&list=PLgZuj5WdK69GiFlqdgo04LvV7RdiEVg_F

1015. "Butterfly Cartoon #8129", *Andertoons* – https://andertoons.com/butterfly/cartoon/8129/flap-wings-crazy-things-caused-sounds-lot-responsibility

1016. "VC Fund Returns Are More Skewed Than You Think", *VC Adventure* – https://sethlevine.com/archives/2020/10/vc-fund-returns-are-more-skewed-than-you-think.html

1017. "Why Only 12% of VCs Can Be Considered Successful. Max." by Jason Lemkin, *SaaStr* – https://www.saastr.com/why-only-12-of-vcs-can-be-considered-successful-max/

1018. *Twitter* – https://x.com/thepatwalls/status/1801613247386124746

1019. "9 SaaS differentiation principles: Why most fail – and how to get it right" by Ton Dobbe, *Value Inspiration* – https://valueinspiration.com/saas-differentiation-principles/

1020. "Profit Whale Curve", *YouTube* – https://www.youtube.com/watch?v=g9-TVWNu4Aw&feature=youtu.be

1021. "SaaS Metrics 2.0 – A Guide to Measuring and Improving what Matters" by David Skok, *For Entrepreneurs* – https://web.archive.org/web/20130206135158/http://www.forentrepreneurs.com/saas-metrics-2/

1022. "David Skok, GP @ Matrix Partners: Driving SaaS Success Using Key Metrics (Video + Transcript)" by Gretchen DeKnikker, *SaaStr* – https://www.saastr.com/david-skok-gp-matrix-partners-driving-saas-success-using-key-metrics-video-transcript/

1023. "28 Customer Onboarding Stats to Know" by Clint Fontanella, *HubSpot* – https://blog.hubspot.com/service/customer-onboarding-stats

1024. "Mystery of a broken A/B test", *YouTube* – https://www.youtube.com/watch?v=FaUO2-AQmr0&feature=youtu.be

1025. "Thanksgiving Cartoon #704", *Andertoons* – https://andertoons.com/thanksgiving/cartoon/704/no-i-dont-want-to-make-wish-where-did-you-get-that

1026. *Twitter* – https://x.com/petergyang/status/1482756170825666562?s=20

1027. "Sales Cartoon #8580", *Andertoons* – https://andertoons.com/sales/cartoon/8580/absolutely-100-percent-guaranteed-most-of-the-time

1028. "Central limit theorem", *Wikipedia* – https://en.wikipedia.org/wiki/Central_limit_theorem

1029. "Lindeberg's condition", *Wikipedia* – https://en.wikipedia.org/wiki/Lindeberg%27s_condition

1030. "VC Fund Returns Are More Skewed Than You Think", *VC Adventure* – https://sethlevine.com/archives/2020/10/vc-fund-returns-are-more-skewed-than-you-think.html

1031. "Why Only 12% of VCs Can Be Considered Successful. Max." by Jason Lemkin, *SaaStr* – https://www.saastr.com/why-only-12-of-vcs-can-be-considered-successful-max/

1032. "Profits Cartoon #7790", *Andertoons* – https://andertoons.com/profits/cartoon/7790/profits-excess-revenues-period-of-time-it-doesnt-look-good

1033. "Buffer: Social media management for everyone", *Buffer: All-you-need social media toolkit for small businesses* – https://buffer.com/

1034. "Transparent Metrics Dashboard", *Buffer: All-you-need social media toolkit for small businesses* – https://buffer.com/metrics

1035. *Twitter* – https://x.com/joelgascoigne?lang=en

1036. "2023 Annual Shareholder Letter" by Joel GascoigneFounder CEO, Bufferjoel.start.page, *Buffer: All-you-need social media toolkit for small businesses* – https://buffer.com/shareholders/2023

1037. "Meritech: The Average SaaS Company IPO's at $225m in ARR. And is No Longer in the Bay Area." by Jason Lemkin, *SaaStr* – https://www.saastr.com/meritech-the-average-saas-company-ipos-at-225m-in-arr/

1038. "Quick ratio", *Wikipedia* – https://en.wikipedia.org/wiki/Quick_ratio

1039. "Work Cartoon #2714", *Andertoons* – https://andertoons.com/work/cartoon/2714/youre-right-its-not-much-of-mission-statement?searchid=44336

1040. "Why Your Mission And Vision Statements Don't Work (And What To Do About It)" by Jeroen Kraaijenbrink, *Forbes* – https://www.forbes.com/sites/jeroenkraaijenbrink/2021/04/13/why-your-mission-and-vision-statements-dont-work-and-what-to-do-about-it/?sh=302415d934e2

1041. "Our Mission", *Patagonia Hong Kong Online* – https://www.patagonia.com.hk/pages/our-mission

1042. "Tesla - ZETA", *logo* – https://www.zeta.org/members/tesla

1043. "Program Partners", *Leading for Life Logo* – https://www.leadingforlife.org/apps/pages/index.jsp?uREC_ID=215122&type=d&termREC_ID=&pREC_ID=622328

1044. https://www.mcdonalds.com/content/dam/sites/uk/nfl/pdf/corporate-governance-framework-2020.pdf

1045. "FedEx Mission Statement" by Daniel Pereira, *Business Model Analyst* – https://businessmodelanalyst.com/fedex-mission-statement/

1046. "Khan Academy" – https://www.khanacademy.org/about

1047. "Our Story", *TOMS* – https://www.toms.com/en-us/about-toms?srsltid=AfmBOorbm-AZgcpmDNb-CaomWWqBynd1mIXcTrpyzyYJXlq8m-QJ2LecU

1048. "Supply Chain Transparency", *TOMS* – https://www.toms.com/en-us/supply-chain-transparency?srsltid=AfmBOoq2xrheJn8FqrUd0o6IgsaVRrpxtm-zlucJOal2MET0bVAabi4m

1049. "About", *DuckDuckGo* – https://duckduckgo.com/about

1050. *Amazon* – https://www.amazon.com/Start-Why-Leaders-Inspire-Everyone/dp/1591846447/ref=asc_df_1591846447?mcid=9a964f9a153333f2b3dac910f6834f16&hvocijid=6120324168534521092-1591846447-&hvexpln=73&tag=hyprod-20&linkCode=df0&hvadid=721245378154&hvpos=&hvnetw=g&hvrand=6120324168534521092&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9028297&hvtargid=pla-2281435176458&psc=1

1051. "How great leaders inspire action" by Simon Sinek, *TED Talks* – https://www.ted.com/talks/simon_sinek_how_great_leaders_inspire_action?language=en

1052. "The Golden Circle", *Simon Sinek* – https://simonsinek.com/golden-circle/

1053. "Corporate Purpose and Financial Performance" by See all articles by Claudine Madras Gartenberg – https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2840005

1054. https://www.munich-business-school.de/fileadmin/user_upload/forschung/working_papers/MBS-WP-2022-05_Antonio_Schmidkonz_Kraft_Purpose_Washing.pdf

1055. https://www.scirp.org/journal/paperinformation?paperid=101900

1056. "For fast and secure sites", *Jamstack.org* – https://jamstack.org/

1057. "The Secret Tesla Motors Master Plan (just between you and me)", *Tesla* – https://www.tesla.com/secret-master-plan

1058. https://en.wikipedia.org/wiki/Patagonia,_Inc.

1059. "Interview Cartoon #7462", *Andertoons* – https://andertoons.com/interview/cartoon/7462/a-doer-get-results-make-things-happen-intentionally-vague

1060. "Business Cartoon #5966", *Andertoons* – https://andertoons.com/business/cartoon/5966/i-dont-get-it-weve-got-mission-statement-credo-and-mantra

1061. "24 Meaningful Facebook Statistics For Marketers In 2024" by Alex Chris, *reliablesoft.net* – https://www.reliablesoft.net/facebook-statistics/

1062. "Buffer: Social media management for everyone", *Buffer: All-you-need social media toolkit for small businesses* – https://buffer.com/

1063. "Ranking Programming Languages by GitHub Users" by Ben Frederickson – https://www.benfrederickson.com/ranking-programming-languages-by-github-users/

1064. "What Percentage of Internet Traffic Is Mobile? [Oct '24 Upd]", *Oberlo* – https://www.oberlo.com/statistics/mobile-internet-traffic

1065. Laura Roeder, *littlefish* – https://lauraroeder.com/?gi=4dfb6a08ae4a

1066. "Transparent Metrics Dashboard", *Buffer: All-you-need social media toolkit for small businesses* – https://buffer.com/metrics

1067. "How HubSpot "Fixed" Retention" by Brian Halligan – https://www.linkedin.com/pulse/how-hubspot-fixed-retention-brian-halligan-rbxce/

1068. *Twitter* – https://x.com/Patticus/status/1702313260547006942

1069. https://twitter.com/Timb03/status/1843073740156100858

1070. *Twitter* – https://x.com/jackfriks/status/1847240738393755756

1071. "Zendesk Apologizes For Suddenly Hiking Its Prices, CEO Hopes For "Make-Up Sex"" by Robin Wauters, *TechCrunch* – https://techcrunch.com/2010/05/20/zendesk-apologizes-for-suddenly-hiking-its-prices-ceo-hopes-for-make-up-sex/

1072. "Dropbox" by Craig Smith, *DMR* – https://expandedramblings.com/index.php/dropbox-statistics/

1073. "Forecast Cartoon #6965", *Andertoons* – https://andertoons.com/forecast/cartoon/6965/if-you-extend-out-5-years-we-own-around-400-of-the-market

1074. *Twitter* – https://x.com/volodarik

1075. "Find and Hire Developers for Startups – Lemon.io – Top 1% Software Developers", *Lemon.io* – https://lemon.io/

1076. *Twitter* – https://x.com/volodarik/status/1846603494289207427

1077. *Twitter* – https://x.com/jlogic

1078. "Christmas Cartoon #7423", *Andertoons* – https://andertoons.com/Christmas/cartoon/7423/naughty-children-who-dont-get-any-toys-an-underserved-niche

1079. "The mid-market briar patch" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/mid-market/

1080. "Rebrand Cartoon #7583", *Andertoons* – https://andertoons.com/rebrand/cartoon/7583/transitioning-from-small-medium-busines-focus-tall-grande

1081. "Fantasy Cartoon #8797", *Andertoons* – https://andertoons.com/fantasy/cartoon/8797/fabled-sword-magic-free-cupcake

1082. "Slate Gray Ribbed Knit Wool And Cashmere Cardigan", *MiuMiu* – https://www.miumiu.com/ie/en/p/ribbed-knit-wool-and-cashmere-cardigan/MMF929_150H_F0480_S_OOO

1083. https://lg.provenpixel.com/ck.php?oaparams=2__bannerid=190907__zoneid=7795__OXLCA=1__cb=9k175e8b2f__x_udomain=tinyurl.com__x_pid3=v-3045__vtag=K:1;F:0.02;D:llbean.com;P:48b100;I:68508440-aaf9-42aa-b51c-55da298220a8;R:u;H:i16-5-168;T:95;__z=divvgHOs7Tdb9ZEwsoaYHpjjenvtqzb9XxANEf9P9MgHYNlt-giy7ZqThgJyWopU8MWDMChCYk__req.clickid=mbzkn2lffd0002cl0o4nk__reqDestUrl=https%3A%2F%2Fglobal.llbean.com%2Fllb%2Fshop%2F126183.html%3Fcgid%3D594%26page%3Dmens-beans-heritage-soft-cotton-fisherman-crew-mens-tall__oadest=https%3A%2F%2Fus.themediaclicks.com%2Flinks%2FmsmRZ50dnRri-UHY54B9s96Vr%3Fts%3DSK%26tsAcc%3DBS%26sub-Id%3Ds2a818b0646682f5%26ramdom%3D9k175e8b2f%26clickId%3Dd28ku-ZIpNKgH7UQAJBUX6CSl9xMjYKFoHqT75bgJBc5FNFqUxGLIq7f4umxi-Bl0kZeE3Sdw3WwpiV1gJqo4UgitYfI8XS%26deviceIp%3D72.177.0.241%26time-stamp%3D1750107200.8796%26campaignId%3D304647%26device-City%3DAustin%26campaignCpc%3D0.05%26landingPage%3Dhttps%253A%252F%252Fglobal.llbean.com%252Fllb%252Fshop%252F126183.html%253Fcgid%253D594%2526page%253Dmens-beans-heritage-soft-cotton-fisherman-crew-mens-tall%26trafficType%3D6%26deviceRegion%3DTX%26device-Country%3DUSA%26deviceUserAgent%3Dnode

1084. "Redirecting..." – https://www.shopping123.com/mbdeals?zid=3w6aLRYHBhmfYA

1085. "Fast Fashion Ad - SNL", *YouTube* – https://www.youtube.com/watch?v=MKTN2OiR2R8&ab_channel=SaturdayNightLive

1086. "Cover your ass", *Wikipedia* – https://en.wikipedia.org/wiki/Cover_your_ass

1087. "Apparel Market Size to Grow USD 2.97 trillion by 2033" by Straits Research – https://straitsresearch.com/report/apparel-market

1088. "Luxury Fashion Market Size, Share, Trends & Forecast 2033" – https://www.imarcgroup.com/luxury-fashion-market

1089. "Fast Fashion Market Report 2025 - Research and Markets" by Research and Markets ltd, *Logo Research and Markets* – https://www.researchandmarkets.com/reports/5939667/fast-fashion-market-report

1090. "L.L.Bean Announces 2023 Year-End Results and Employee Contributions", *L.L.Bean* – https://www.llbean.com/llb/shop/518872?page=llbean-announces-2023-year-end-results-and-employee-contributions

1091. "Mastering Profit Margins: A Comprehensive Guide for Clothing Retailers" by Tristan Sigerson, *New Frontier Funding* – https://newfrontierfunding.com/understanding-cost-based-pricing-apparel-business/

1092. https://www.pradagroup.com/content/dam/pradagroup/documents/investors/FY-2024/Prada_Group_FY24_PR.pdf

1093. "Intentional Weakness: Create an uncopyable strategy like these companies", *YouTube* – https://www.youtube.com/watch?v=dVSXNkWUq88&ab_channel=JasonCohen

1094. "Wine Cartoon #305", *Andertoons* – https://andertoons.com/wine/cartoon/305/and-for-connoisseur-on-budget-we-have-lovely-tuesday-vintage-at-buck-fifty

1095. "Barking Cartoon #7939", *Andertoons* – https://andertoons.com/barking/cartoon/7939/buy-sell-sell-buy-bark-woof-sorry-buy

1096. "WhatsUp Gold", *Other Features* – https://web.archive.org/web/20001206213700/http://www.whatsupgold.com/Products/WhatsUp/features.html

1097. "HP OpenView" – https://web.archive.org/web/20000511100228/http://www.openview.hp.com/

1098. "Business Cartoon #8527", *Andertoons* – https://andertoons.com/business/cartoon/8527/lots-of-acronyms-acrostic-help-remember

1099. "The Magical Number Seven, Plus or Minus Two", *Wikipedia* – https://en.wikipedia.org/wiki/The_Magical_Number_Seven,_Plus_or_Minus_Two

1100. "Card Cartoon #9215", *Andertoons* — https://andertoons.com/card/cartoon/9215/where-do-see-myself-five-years

1101. "Weather Cartoon #5221", *Andertoons* — https://andertoons.com/weather/cartoon/5221/but-to-be-fair-theres-fifty-percent-chance-of-just-about-anything

1102. "Brand awareness", *Wikipedia* — https://en.wikipedia.org/wiki/Brand_awareness

1103. "Rebrand Cartoon #7618", *Andertoons* — https://andertoons.com/rebrand/cartoon/7618/instead-failure-rebrand-dark-gritty-reboot-previous-profits

1104. *The idea maze* — https://cdixon.org/2013/08/04/the-idea-maze/

1105. "Education Cartoon #6782", *Andertoons* — https://andertoons.com/education/cartoon/6782/i-know-its-wrong-im-just-waiting-for-the-autocorrect

1106. "Responsibility Cartoon #7817", *Andertoons* — https://andertoons.com/responsibility/cartoon/7817/dont-care-whose-time-it-is-were-going-to-have-a-nice-dinner

1107. "How big was the first iPhone? How the iPhone's screen size has changed over time", *Mic* — https://www.mic.com/articles/164769/how-big-was-the-first-i-phone-how-the-i-phone-s-screen-size-has-changed-over-time

1108. "The Twelve-Factor App" by Adam Wiggins — https://www.12factor.net/

1109. "Signal v. Noise is closed.", *SvN* — https://signalvnoise.com/

1110. "Death Cartoon #1705", *Andertoons* — https://andertoons.com/death/cartoon/1705/any-way-we-can-hurry-this-up

1111. "Stock Cartoon #710", *Andertoons* — https://andertoons.com/stock/cartoon/710/stocks-rose-slightly-in-early-trading-then-plummeted-on-news-that-stocks-rose-slightly-in-early-trading

1112. "The right way to position against competition" by Jason, *@ASmartBear* — https://blog.asmartbear.com/competitive-positioning/

1113. "Graph Cartoon #7231", *Andertoons* — https://andertoons.com/graph/cartoon/7231/serendipity-is-up-fluke-doing-well-concerned-about-dumb-luck

1114. "Solving the "marketplace" business model" by Jason, *@ASmartBear* — https://blog.asmartbear.com/marketplace-business-model/

1115. "Time Cartoon #7579", *Andertoons* — https://andertoons.com/time/cartoon/7579/behind-schedule-on-time-machine-catch-up-once-finished

1116. "Spanner: TrueTime and external consistency", *Google Cloud* — https://cloud.google.com/spanner/docs/true-time-external-consistency

1117. "Vector clock", *Wikipedia* — https://en.wikipedia.org/wiki/Vector_clock

1118. https://stackoverflow.com/questions/4770635/s3-error-the-difference-between-the-request-time-and-the-current-time-is-too-la

1119. https://cse.buffalo.edu/tech-reports/2014-04.pdf

1120. https://twitter.com/akannett/status/605430520474927105

1121. "Should you invent a new UX?" by Jason, *@ASmartBear* — https://blog.asmartbear.com/new-ux/

1122. "Helping your customers navigate change" by Beau Ulrey, *UX Collective* — https://uxdesign.cc/help-your-customer-navigate-change-1bb674f6ce23?gi=b4127f9f7f30

1123. "Startup Best Practices 8 -Getting the Most from Your Team By Preventing Burnout by @ttunguz" — https://tomtunguz.com/managing-energy/

1124. "On the value of Judgement" by Jason Cohen, *A Smart Bear* — https://longform.asmartbear.com/judgement/

1125. "Advertising Cartoon #106", *Andertoons* — https://andertoons.com/advertising/cartoon/106/its-not-that-i-dont-like-you-ted-you-just-dont-fit-my-target-demographic

1126. "Idiom Cartoon #7414", *Andertoons* — https://andertoons.com/idiom/cartoon/7414/so-which-whole-do-we-want-to-go-with

1127. "Marketing Platform Independence" by Jason Cohen, *A Smart Bear* — https://longform.asmartbear.com/marketing-platform-independence/

1128. "The full story of "the one important thing" for startups" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/one-priority/

1129. "Decision Cartoon #6618", *Andertoons* – https://andertoons.com/decision/cartoon/6618/do-it-dont-do-it-we-dont-care-anymore

1130. "News Cartoon #4033", *Andertoons* – https://andertoons.com/news/cartoon/4033/reaction-to-news-was-mixed-largely-because-we-asked-more-than-one-person

1131. "Internet Cartoon #6232", *Andertoons* – https://andertoons.com/internet/cartoon/6232/i-had-whale-dream-again

1132. https://twitter.com/mc_hankins/status/653621441053753344

1133. "Weather Cartoon #6825", *Andertoons* – https://andertoons.com/weather/cartoon/6825/and-now-the-7-day-forecast

1134. "Weather Cartoon #5221", *Andertoons* – https://andertoons.com/weather/cartoon/5221/but-to-be-fair-theres-fifty-percent-chance-of-just-about-anything

1135. "Sales Cartoon #6106", *Andertoons* – https://andertoons.com/sales/cartoon/6106/seriously-no-one-saw-this-coming

1136. https://www.antofagasta.climatemps.com/precipitation.php

1137. https://longform.asmartbear.com/ref/Brier-1950.pdf

1138. "Brier score", *Wikipedia* – https://en.wikipedia.org/wiki/Brier_score

1139. https://longform.asmartbear.com/ref/Steins-Paradox-by-Efron-and-Morris.pdf

1140. "Evolution Cartoon #1914", *Andertoons* – https://andertoons.com/evolution/cartoon/1914/its-fad

1141. "Darwinian advice for "Going Viral"" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/going-viral/

1142. "Batman Cartoon #9068", *Andertoons* – https://andertoons.com/batman/cartoon/9068/shareholder-value-justice

1143. "Buffer: Social media management for everyone", *Buffer: All-you-need social media toolkit for small businesses* – https://buffer.com/

1144. "Introducing Buffer's Salary Calculator & New Salary Formula" by Joel and Leo, Buffer Co-founders, *Buffer: All-you-need social media toolkit for small businesses* – https://buffer.com/resources/transparent-salaries/

1145. "Shoes: Fashionable Footwear for Every Step", *TOMS* – https://www.toms.com/en-us

1146. "Enable Cookies", *USAA* – https://www.usaa.com/?akredirect=true

1147. "Business Cartoon #2802", *Andertoons* – https://andertoons.com/business/cartoon/2802/im-sorry-ted-i-think-we-need-to-slow-down-ive-been-hurt-in-mergers-before

1148. "Pando: Democratizing career progression" by Mark FreinChief Workplace Officer, Oyster, *Oyster logo* – https://www.pando.com/new

1149. "Business Cartoon #2315", *Andertoons* – https://andertoons.com/business/cartoon/2315/but-to-be-fair-theres-no-business-like-show-business

1150. https://twitter.com/ericjackson/status/400972956689367041

1151. "Fruit Cartoon #5516", *Andertoons* – https://andertoons.com/fruit/cartoon/5516/explain-to-me-how-comparing-apples-and-oranges-fruitless

1152. "The Trap of Relative Value" by Brad Feld, *Feld Thoughts* – https://feld.com/archives/2014/10/trap-relative-value/

1153. "Selling Cartoon #6497", *Andertoons* – https://andertoons.com/selling/cartoon/6497/idea-not-only-grow-piece-of-pie-but-expand-into-other-pastries

1154. "Haiku error messages - GNU Project", *Free Software Foundation* – https://www.gnu.org/fun/jokes/error-haiku.html

1155. "Devil Cartoon #6064", *Andertoons* – https://andertoons.com/devil/cartoon/6064/boy-what-day-huh

1156. "Reindeer Cartoon #7421", *Andertoons* – https://andertoons.com/reindeer/cartoon/7421/i-understand-reindeer-game-but-watching-other-reindeer-play

1157. "The Three-Sentence Email" by Manu Ebert, *Medium* – https://medium.com/@maebert/how-to-write-emails-like-a-grown-up-4952ffb002dd

1158. "Forecast Cartoon #6965", *Andertoons* – https://andertoons.com/forecast/cartoon/6965/if-you-extend-out-5-years-we-own-around-400-of-the-market

1159. "SaaS Metrics 2.0 - A Guide to Measuring and Improving what Matters" by David Skok, *For Entrepreneurs* – https://www.forentrepreneurs.com/saas-metrics-2/

1160. "Fool Cartoon #6048", *Andertoons* – https://andertoons.com/fool/cartoon/6048/so-you-gonna-rush-in-or-what

1161. "Therapy Cartoon #5232", *Andertoons* – https://andertoons.com/therapy/cartoon/5232/i-used-to-have-lot-of-issues-but-then-i-just-canceled-my-subscription

1162. "5 Interesting Learnings from Fastly at $500,000,000+ in ARR" by Jason Lemkin, *SaaStr* – https://www.saastr.com/5-interesting-learnings-from-fastly-at-50000000-in-arr/

1163. "Food Cartoon #2917", *Andertoons* – https://andertoons.com/food/cartoon/2917/no-youre-right-eat-better

1164. "Out of the cesspool and into the sewer: A/B testing trap" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/local-minimum/

1165. "Mystery of a broken A/B test", *YouTube* – https://www.youtube.com/watch?v=FaUO2-AQmr0&feature=youtu.be

1166. "Fortune Cartoon #6108", *Andertoons* – https://andertoons.com/fortune/cartoon/6108/i-see-big-changes-in-your-future-which-reminds-me-my-fee-going-up-beginning-next-week

1167. "Home", *Capital Factory* – https://capitalfactory.com/

1168. "The truth about conversion ratios for downloadable software" by Andy Brice, *Successful Software* – https://successfulsoftware.net/2009/04/23/the-truth-about-conversion-ratios-for-software/

1169. "Psychology Cartoon #6178", *Andertoons* – https://andertoons.com/psychology/cartoon/6178/by-way-im-having-special-this-month-have-one-breakthrough-get-another-free

1170. "Dropbox Hits 100 Million Users Says Drew Houston" by Victoria Barret, *Forbes* – https://www.forbes.com/sites/victoriabarret/2012/11/13/dropbox-hits-100-million-users-says-drew-houston/

1171. *Twitter* – https://x.com/StartupArchive_/status/1853781815821287845

1172. "The truth about conversion ratios for downloadable software" by Andy Brice, *Successful Software* – https://successfulsoftware.net/2009/04/23/the-truth-about-conversion-ratios-for-software/

1173. "Understand Your SaaS Free Trial Conversion Rates Numbers, Uncover Insights and Money" by Aiza Domingcil, *Encharge* – https://encharge.io/saas-free-trial-conversion-rates/

1174. "SaaS Free Trial Conversion Rate Benchmarks" by Evan Bailyn, *First Page Sage* – https://firstpagesage.com/seo-blog/saas-free-trial-conversion-rate-benchmarks/

1175. "Darwinian advice for "Going Viral"" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/going-viral/

1176. "Zero-sum marketing channels: Good or bad for a startup to pursue?" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/zero-sum-marketing/

1177. "Secrets Of Freemium Pricing: Make The Cheapskates Pay" by Dharmesh Shah, *OnStartups* – https://www.onstartups.com/tabid/3339/bid/37737/Secrets-Of-Freemium-Pricing-Make-The-Cheapskates-Pay.aspx

1178. "How to create a profitable Freemium startup (spreadsheet model included!) at andrewchen" – https://andrewchen.com/how-to-create-a-profitable-freemium-startup-spreadsheet-model-included/

1179. "Three freemium strategies" – https://www.startuplessonslearned.com/2009/01/three-freemium-strategies.html

1180. "Death Cartoon #7973", *Andertoons* – https://andertoons.com/death/cartoon/7973/oh-thank-god-stacy-its-for-you

1181. "Pascal's wager", *Wikipedia* – https://en.wikipedia.org/wiki/Pascal%27s_wager

1182. "Hello, I'm 1074018628" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/hello-support/

1183. "How Perfect Pricing got me 1500 Sales in 2 Days" by wpengine, *@ASmartBear* – https://blog.asmartbear.com/perfect-pricing/

1184. "Perfect Pricing Part Deux - More money from fewer sales" by Jason, *@ASmartBear* – https://blog.asmartbear.com/higher-pricing/

1185. "Music Cartoon #6016", *Andertoons* – https://andertoons.com/music/cartoon/6016/i-think-if-more-french-horn-were-going-to-work-it-would-have-by-now

1186. "Tv Cartoon #5713", *Andertoons* – https://andertoons.com/TV/cartoon/5713/only-picture-high-definition-shows-are-as-bad-as-always

1187. "Starting Your Own Business with Erica Douglass - Erica Douglass, "temporarily retired" after selling a successful business at age 26, writes thought-provoking blog entries challenging you to change your life and daring you to become more successful.", *Starting Your Own Business with Erica Douglass* – https://www.erica.biz/

1188. "The Nastiest Habit (That We All Have)" by Lisa Smith, *Starting Your Own Business with Erica Douglass* – https://www.erica.biz/2012/the-nastiest-habit/

1189. "High-concept pitches are not your friend" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/high-concept-pitch/

1190. "Economy Cartoon #3502", *Andertoons* – https://andertoons.com/economy/cartoon/3502/due-to-recent-economic-conditions-picture-worth-has-dropped-to-an-all-time-low-of-842-words

1191. "The full story of "the one important thing" for startups" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/one-priority/

1192. "Insect Cartoon #4408", *Andertoons* – https://andertoons.com/insect/cartoon/4408/so-where-do-you-see-yourself-in-ten-minutes

1193. "Strong opinions, somewhat weakly held" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/strong-opinions-weakly-held/

1194. "Let's just call plans what they are: guesses" by Amber, *Signal v. Noise by Basecamp* – https://signalvnoise.com/posts/1805-lets-just-call-plans-what-they-are-guesses

1195. "Benefits Cartoon #7693", *Andertoons* – https://andertoons.com/benefits/cartoon/7693/competitive-salary-vacation-and-pokemon-around-the-office

1196. "WP Engine Careers - Be you. Be here." by Chairwoman and CEO, WP Engine, *WP Engine Careers* – https://wpengine.careers/

1197. https://twitter.com/bmenell

1198. "Do I dare call bullshit aloud? Do you?" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/call-bullshit/

1199. "How to Write a Cover Letter that Gets Read – Work Awesome" by Jason Cohen – https://workawesome.com/your-job/how-to-write-a-cover-letter-that-gets-read/

1200. "WP Engine, the Smarter Way to Host WordPress", *WP Engine* – https://wpengine.com/

1201. https://otherinbox.com/

1202. https://www.austinonrails.org/

1203. https://www.lonestarrubyconf.com/

1204. "Build, Measure, Learn cycle" by Dominic Rogers, *Medium* – https://medium.com/@dominic_11011/build-measure-learn-cycle-ace388a13b4d

1205. "Lessons Learned" – https://www.startuplessonslearned.com/

1206. *Amazon* – https://www.amazon.com/gp/product/0307887898/ref=as_li_ss_tl?ie=UTF8&tag=asmbe-20&linkCode=as2&camp=217145&creative=399373&creativeASIN=0307887898

1207. "Psychiatry Cartoon #1114", *Andertoons* – https://andertoons.com/psychiatry/cartoon/1114/i-told-you-its-an-inkblot-theyre-all-inkblots

1208. "Home", *Capital Factory* – https://capitalfactory.com/

1209. "The three stages of expertise ..." by swardley – https://blog.gardeviance.org/2008/04/three-stages-of-expertise.html

1210. "LGBTQ people", *Wikipedia* – https://en.wikipedia.org/wiki/LGBT

1211. "Procrastinate for Success!" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/procrastinate/

1212. *Amazon* – https://www.amazon.com/gp/product/0142000280/ref=as_li_ss_tl?ie=UTF8&tag=asmbe-20&linkCode=as2&camp=217145&creative=399349&creativeASIN=0142000280

1213. https://www.43folders.com/izero

1214. "How I got 6000 RSS subscribers in 12 months" by Jason, *@ASmartBear* – https://blog.asmartbear.com/how-i-got-6000-rss-subscribers-in-12-months/

1215. "A tour of my Wordpress plugins" by Jason, *@ASmartBear* – https://blog.asmartbear.com/wordpress-plugins/

1216. *The Adam Carolla Show* – https://adamcarolla.com/

1217. "Jason Fried" – https://world.hey.com/jason

1218. "Learn from Proven Entrepreneurs", *Business Podcast for Startups* – https://mixergy.com/

1219. "Turning The Tables On Me - with Jason Cohen", *Business Podcast for Startups* – https://mixergy.com/interviews/jason-cohen-wpengine-interview/

1220. "From Burn-Out to $100M in ARR with Jason Cohen, Founder of WP Engine", *YouTube* – https://www.youtube.com/watch?v=Kn5MbZoSRNM&feature=youtu.be

1221. "You're a little company, now act like one" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/little-company/

1222. "Word Cartoon #5005", *Andertoons* – https://andertoons.com/word/cartoon/5005/were-really-more-of-department

1223. https://twitter.com/chrisbrogan

1224. https://twitter.com/ZAPPOS

1225. AmazonAmazon, *Chris Brogan* – https://chrisbrogan.com/

1226. "HubSpot Blog" by Maxwell Iskiev, *HubSpot* – https://blog.hubspot.com/

1227. "If Kindergarten were like Social Media Marketing" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/social-media-marketing/

1228. https://ittybiz.s3.amazonaws.com/freebievault/IttyBiz_Back_In_The_Day.pdf

1229. "Painless Functional Specifications – Part 4: Tips" by Joel Spolsky, *Joel on Software* – https://www.joelonsoftware.com/2000/10/15/painless-functional-specifications-part-4-tips/

1230. "Don't Follow Your Passion" by Amy Hoy, *Stacking the Bricks* – https://stackingthebricks.com/dont-follow-your-passion/

1231. "Politics Cartoon #4818", *Andertoons* – https://andertoons.com/politics/cartoon/4818/mom-and-dad-are-my-primary-funding-but-i-get-plenty-of-soft-money-from-my-grandparents

1232. "Zero-sum marketing channels: Good or bad for a startup to pursue?" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/zero-sum-marketing/

1233. "Too small to fail: How startups can grow in recessions" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/startups-in-recessions/

1234. "Family Cartoon #1009", *Andertoons* – https://andertoons.com/family/cartoon/1009/you-know-for-just-small-increase-in-my-allowance-i-could-make-bedtime-lot-easier

1235. "Therapy Cartoon #5952", *Andertoons* – https://andertoons.com/therapy/cartoon/5952/the-fame-fine-and-moneys-great-but-sometimes-i-miss-plumbing

1236. "WP Engine, the Smarter Way to Host WordPress", *WP Engine* – https://wpengine.com/

1237. https://www.ericsink.com/

1238. https://www.ericsink.com/bos/Micro_ISV.html

1239. https://www.ericsink.com/bos/First_MicroISV_Report.html

1240. *Twitter* – https://x.com/damengchen/status/1755333144532967471?s=20

1241. *Twitter* – https://x.com/TomBilyeu/status/1853180382201405770

1242. https://twitter.com/robwalling/status/1755305197088227712?s=20

1243. "Don't Build the SaaS Around the Audience, Build the Audience Around the SaaS" by Laura Roeder, *littlefish* – https://lauraroeder.com/dont-build-the-saas-around-the-audience-build-the-audience-around-the-saas-82519b1ad6e5?gi=001c3c9a8e19

1244. "Startup Weekend", *Techstars Startup Events* – https://www.techstars.com/communities/startup-weekend

1245. https://twitter.com/aaronbrazell

1246. *Amazon* – https://www.amazon.com/gp/product/0470568135?ie=UTF8&tag=asmbe-20&linkCode=as2&camp=1789&creative=390957&creativeASIN=0470568135

1247. https://www.ericsink.com/entries/wsol_sold.html

1248. "Cicada", *Wikipedia* – https://en.wikipedia.org/wiki/Cicada

1249. "Standing out from the noise" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/standing-out-from-the-noise/

1250. "Balsamiq: Fast, focused wireframing tools" – https://balsamiq.com/

1251. "Hit $100,000 in revenue, time to start looking up" by Peldi Guilizzoni – https://balsamiq.com/blog/hit-100000-in-revenue-time-to-start-looking-up/

1252. "Balsamiq: week one recap" by Peldi Guilizzoni – https://balsamiq.com/blog/balsamiq-week-one-recap/

1253. "The Balsamiq Software Donation Program" by Peldi Guilizzoni – https://balsamiq.com/blog/donating/

1254. "Balsamiq roadmap for 2010" by Peldi Guilizzoni – https://balsamiq.com/blog/balsamiq-roadmap-for-2010/

1255. "Balsamiq: week one recap" by Peldi Guilizzoni – https://balsamiq.com/blog/balsamiq-week-one-recap/

1256. "The Balsamiq Playbook" – https://balsamiq.com/company/playbook/

1257. "Free Online Appointment Scheduling Software", *Calendly.com* – https://calendly.com/

1258. "Mutual availability overview", *Help Center* – https://help.calendly.com/hc/en-us/articles/360040973354-Mutual-availability-overview

1259. "Free Online Scheduling Tool - YouCanBookMe", *See all integrations* – https://youcanbook.me/

1260. "Scheduling Software Everyone Will Love · SavvyCal" – https://savvycal.com/

1261. "Darwinian advice for "Going Viral"" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/going-viral/

1262. "You're a little company, now act like one" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/little-company/

1263. "How to bring a product to market / A very rare interview with Sean Ellis" by Nivi, *Venture Hacks* – https://venturehacks.com/sean-ellis-interview

1264. "How to bring a product to market, Part 2 – after product/market fit" by Nivi, *Venture Hacks* – https://venturehacks.com/sean-ellis-interview-2

1265. https://vcwear.com/

1266. "Home", *Capital Factory* – https://capitalfactory.com/

1267. "Light cone", *Wikipedia* – https://en.wikipedia.org/wiki/Light_cone

1268. https://evhead.com/2005/11/ten-rules-for-web-startups.asp

1269. "ROYAL188: Situs Slot Login Online Cocok Untuk Slotter Penikmat Hotel Don Candido" by ROYAL188 – https://hoteldoncandido.com/

1270. "Case Study: Using an LOI to get customer feedback on a minimum viable product" – https://www.startuplessonslearned.com/2009/10/case-study-using-loi-to-get-customer.html

1271. "10 things I've never heard a successful startup founder say" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/successful-founders-dont-say/

1272. "You're a little company, now act like one" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/little-company/

1273. https://blendingthemix.com/2009/01/23/the-most-popular-100-twitter-applications/

1274. "Twitter Fan Wiki / Apps" – http://twitter.pbworks.com/w/page/1779726/Apps

1275. https://failblog.org/

1276. "Christmas Cartoon #4319", *Andertoons* – https://andertoons.com/Christmas/cartoon/4319/i-do-have-solution-but-it-means-moving-to-santas-naughty-list

1277. "Stock Cartoon #4122", *Andertoons* – https://andertoons.com/stock/cartoon/4122/what-happens-to-stock-when-i-do-this

1278. "No, that IS NOT a competitive advantage" by Jason, *@ASmartBear* – https://blog.asmartbear.com/not-competitive-advantage/

1279. "SmartBear Software - The confidence behind your code.", *SmartBear Software* – https://smartbear.com/

1280. Joel Spolsky, *Joel on Software* – https://www.joelonsoftware.com/

1281. "Signal v. Noise is closed.", *SvN* – https://signalvnoise.com/

1282. *Amazon* – https://www.amazon.com/gp/product/0307463745?ie=UTF8&amp%3Btag=asmbe-20&amp%3BlinkCode=as2&amp%3Bcamp=1789&amp%3Bcreative=390957&amp%3BcreativeASIN=0307463745

1283. "Please stop saying social media marketing is free" by Jason, *@ASmartBear* – https://blog.asmartbear.com/please-stop-saying-social-media-marketing-is-free/

1284. "How I got 6000 RSS subscribers in 12 months" by Jason, *@ASmartBear* – https://blog.asmartbear.com/how-i-got-6000-rss-subscribers-in-12-months/

1285. "A Tradeshow Checklist, born of experience" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/tradeshow-tips-checklist/

1286. "Best Kept Secrets of Code Review", *Login* – https://smartbear.com/resources/ebooks/best-kept-secrets-of-code-review/

1287. https://itwatchdogs.com/

1288. https://hitenshah.name/

1289. https://kissmetrics.com/

1290. Dave McClure, Practical Venture Capital, *500 Hats* – https://500hats.com/?gi=361e2f910ab0

1291. "About", *Startup Marketing Blog - By Sean Ellis | Unlocking Startup Growth* – https://www.startup-marketing.com/about/

1292. "About the author" – https://www.startuplessonslearned.com/2008/10/about-author.html

1293. https://en.wikiquote.org/wiki/Jerky_Boys

1294. "Discover what's blocking sales with less than a day of work" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/more-sales-customer-feedback/

1295. https://blogs.zoho.com/general/companies-don-t-get-killed-by-competition-they-commit-suicide

1296. "Pick one and own it" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/one-benefit/

1297. "The Better Angels of Our Nature", *Wikipedia* – https://en.wikipedia.org/wiki/The_Better_Angels_of_Our_Nature

1298. "Sports Cartoon #3008", *Andertoons* – https://andertoons.com/sports/cartoon/3008/he-can-play-like-this-but-hell-only-be-able-to-give-105-percent

1299. https://wexler.free.fr/library/files/gilovich%20%281985%29%20the%20hot%20hand%20in%20basketball.%20on%20the%20misperception%20of%20random%20sequences.pdf

1300. "Autocorrelation", *Wikipedia* – https://en.wikipedia.org/wiki/Autocorrelation

1301. "Hot hand", *Wikipedia* – https://en.wikipedia.org/wiki/Hot_hand

1302. "Bible code", *Wikipedia* – https://en.wikipedia.org/wiki/Bible_code

1303. "File:Bible code example.svg - Wikimedia Commons", *Wikipedia* – https://commons.wikimedia.org/wiki/File:Bible_code_example.svg

1304. "There are Codes in War and Peace Too" – https://chance.dartmouth.edu/chance_news/full_text/6.12/CodesinWarPeace.html

1305. "Stock Cartoon #4122", *Andertoons* – https://andertoons.com/stock/cartoon/4122/what-happens-to-stock-when-i-do-this

1306. https://answers.onstartups.com/

1307. "What to do when Social Media creates a Shit Storm" by Kathy Hendershot-Hurd, *Virtual Impax* – https://virtualimpax.com/2009/07/13/social-media-shit-storm/

1308. "SmartBear Software - The confidence behind your code.", *SmartBear Software* – https://smartbear.com/

1309. "Fish Cartoon #5228", *Andertoons* – https://andertoons.com/fish/cartoon/5228/im-looking-for-something-in-small-pond

1310. "Budget Tracker & Planner", *Mint* – https://mint.intuit.com/

1311. "Communicating Values: Show, don't Tell" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/show-dont-tell/

1312. "Meeting Cartoon #512", *Andertoons* – https://andertoons.com/meeting/cartoon/512/so-as-you-can-see-customer-satisfaction-up-considerably-since-phasing-out-complaint-forms

1313. https://twitter.com/einmaleins/statuses/11901946023

1314. https://twitter.com/DishNetwork/statuses/11906944368

1315. "Steve Blank Teaching Entrepreneurship – By Getting Out of the Building" by March 11, 2010 at 6:41 am, *Steve Blank* – https://steveblank.com/2010/03/11/teaching-entrepreneurship-%E2%80%93-by-getting-out-of-the-building/

1316. "Employed with a side of startup" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/working-startup/

1317. "Dog Cartoon #5038", *Andertoons* – https://andertoons.com/dog/cartoon/5038/whats-that-boy-paradigm-shift

1318. "Does anyone else think Word 2007 Sucks? Badly!!" by Pduol, *Straight Dope Message Board* – https://boards.straightdope.com/t/does-anyone-else-think-word-2007-sucks-badly/522372

1319. "When being "first" is not a competitive advantage" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/first-competitive-advantage/

1320. "Web Search Engines History & Evolution", *SEO Consultants* – https://seoconsultants.com/search-engines-history/

1321. *Scott Berkun* – https://scottberkun.com/

1322. "Vista Equity invests in Francisco's SmartBear, pushing valuation to $1.8bn-plus" by Milana Vinn, *PE Hub* – https://www.pehub.com/vista-equity-invests-in-franciscos-smartbear-pushing-valuation-to-1-8bn-plus/?utm_source=longform.asmartbear.com&utm_campaign=longform.asmartbear.com&utm_medium=post

1323. https://itwatchdogs.com/

1324. "Kathy Sierra", *Wikipedia* – https://en.wikipedia.org/wiki/Kathy_Sierra

1325. "minimalism" by goundoulf, *ihumanable* – https://web.archive.org/web/20100408174137/http://ihumanable.com/blog/2009/12/minimalism/

1326. "SmartBear Software - The confidence behind your code.", *SmartBear Software* – https://smartbear.com/

1327. "Butterfly Cartoon #8129", *Andertoons* – https://andertoons.com/butterfly/cartoon/8129/flap-wings-crazy-things-caused-sounds-lot-responsibility

1328. "Butterfly effect", *Wikipedia* – https://en.wikipedia.org/wiki/Butterfly_effect

1329. "Welcome to IttyBiz! We're Glad You're Here.", *IttyBiz* – https://ittybiz.com/

1330. Tim Stoddart, *Copyblogger* – https://copyblogger.com/

1331. https://500hats.typepad.com/

1332. "Urban Dictionary: silent but deadly", *Urban Dictionary* – https://www.urbandictionary.com/define.php?term=silent+but+deadly

1333. "Attractor", *Wikipedia* – https://en.wikipedia.org/wiki/Attractor

1334. "More debunked or questioned psychological studies" by December 31, 2022December 31, 2022, *Why Evolution Is True* – https://whyevolutionistrue.com/2022/12/31/more-debunked-or-questioned-psychological-studies/

1335. "Don't touch when you are shopping, or the new endowment effect" by Tyler Cowen, *Marginal REVOLUTION* – https://marginalrevolution.com/marginalrevolution/2009/01/dont-touch-when-you-are-shopping.html

1336. https://psycnet.apa.org/record/1996-06400-003

1337. https://www.bakadesuyo.com/could-a-single-sneeze-make-you-changes-your-m

1338. "miller-mccune.com" – https://www.miller-mccune.com/health/to-feel-good-reach-for-the-sky-8445/

1339. "Quieting the lizard brain", *Seth's Blog* – https://seths.blog/2010/01/quieting-the-lizard-brain/

1340. https://twitter.com/asmartbear

1341. "How I got 6000 RSS subscribers in 12 months" by Jason, *@ASmartBear* – https://blog.asmartbear.com/how-i-got-6000-rss-subscribers-in-12-months/

1342. "Baby Cartoon #6018", *Andertoons* – https://andertoons.com/baby/cartoon/6018/you-see-any-next-generation-leadership-you-call-me

1343. *Amazon* – https://www.amazon.com/gp/product/0316017922?ie=UTF8&amp%3Btag=asmbe-20&amp%3BlinkCode=as2&amp%3Bcamp=1789&amp%3Bcreative=390957&amp%3BcreativeASIN=0316017922

1344. "Being an expert takes time, not talent" by Penelope, *Penelope Trunk Careers Blog* – https://blog.penelopetrunk.com/2010/01/28/being-an-expert-takes-time-not-talent/

1345. "Freakonomics in the Times Magazine: A Star Is Made", *Freakonomics* – https://freakonomics.com/2006/05/freakonomics-in-the-times-magazine-a-star-is-made/

1346. "Peter Bowen (@petebowen.bsky.social)", *Bluesky Social* – https://bsky.app/profile/petebowen.bsky.social/post/3lqts6sdej22l

1347. "How and Why To Write a Rude Q&A", *Scott Berkun* – https://scottberkun.com/2007/how-to-write-a-rude-qa/

1348. *Andertoons* – https://andertoons.com/search.php?search_query=devil&section=product

1349. "Devil Cartoon #8813", *Andertoons* – https://andertoons.com/devil/cartoon/8813/take-shoes-off-hell?searchid=45102&search_query=devil

1350. "Steve Blank Innovation and Entrepreneurship", *Steve Blank* – https://steveblank.com/

1351. "Ask 37signals: Is it really the number of features that matter?" by Marc, *Signal v. Noise by Basecamp* – https://signalvnoise.com/posts/643-ask-37signals-is-it-really-the-number-of-features-that-matter

1352. https://sethgodin.typepad.com/seths_blog/2008/03/1000-true-fans.html

1353. "The Case Against 1000 True Fans" by Kevin Kelly, *The Technium* – https://kk.org/thetechnium/the-case-agains/

1354. "Chart Cartoon #3709", *Andertoons* – https://andertoons.com/chart/cartoon/3709/things-went-from-bad-to-worse-but-were-hopeful-now-that-were-doing-badly-again

1355. "Product Management, Data & Design Training", *Pragmatic Institute - Corporate* – https://www.pragmaticinstitute.com/

1356. "Steve Johnson on The Strategic Role of Product Management", *Business of Software* – https://businessofsoftware.org/talks/why-software-is-not-a-business/

1357. "The cardinal sin of community management" – https://www.startuplessonslearned.com/2009/09/cardinal-sin-of-community-management.html

1358. "Focused Chaos" by Ben Yoskovitz, *Substack* – https://www.focusedchaos.co/

1359. "A Game For All Ages - TIME", *TIME Logo* – https://web.archive.org/web/20090228020253/http://www.time.com/time/magazine/article/0,9171,1191861-1,00.html

1360. "Steve Jobs does customer development" by Nivi, *Venture Hacks* – https://venturehacks.com/jobs-customer-development

1361. "Discover what's blocking sales with less than a day of work" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/more-sales-customer-feedback/

1362. "Cassette Cartoon #8328", *Andertoons* – https://andertoons.com/cassette/cartoon/8328/cassette-surgery-pencil

1363. "The Essential Guide to Growing Your Blog on Minimal Time" by Darren Rowse, *ProBlogger* – https://problogger.com/the-essential-guide-to-growing-your-blog-on-minimal-time/

1364. https://www.adaringadventure.com/blog/wordpress/featured/imagine-this/

1365. https://www.43folders.com/2005/01/15/patching-your-personal-suck

1366. "Don't wait for retirement to live the good life. Do it now" by Penelope, *Penelope Trunk Careers Blog* – https://blog.penelopetrunk.com/2008/03/20/dont-wait-for-retirement-to-live-the-good-life-do-it-now/

1367. "Keeping up with the Social Web" by jeremiah_owyang, *Jeremiah Owyang* – https://web-strategist.com/blog/2009/05/08/how-do-i-keep-up/

1368. "SharkTank & Success & Motivation", *blog maverick* – https://blogmaverick.com/2007/12/24/success-and-motivation/

1369. "7 Things to consider before launching a startup" by Penelope, *Penelope Trunk Careers Blog* – https://blog.penelopetrunk.com/2009/01/05/7-things-to-consider-before-launching-a-startup/

1370. "Art Cartoon #8563", *Andertoons* – https://andertoons.com/art/cartoon/8563/van-gogh-ear-painting-do-not-attempt-warning

1371. Joel Spolsky, *Joel on Software* – https://www.joelonsoftware.com/

1372. "The Absolute Minimum Every Software Developer Absolutely, Positively Must Know About Unicode and Character Sets (No Excuses!)" by Joel Spolsky, *Joel on Software* – https://www.joelonsoftware.com/2003/10/08/the-absolute-minimum-every-software-developer-absolutely-positively-must-know-about-unicode-and-character-sets-no-excuses/

1373. "Strategy Letter I: Ben and Jerry's vs. Amazon" by Joel Spolsky, *Joel on Software* – https://www.joelonsoftware.com/2000/05/12/strategy-letter-i-ben-and-jerrys-vs-amazon/

1374. "How to Ship Anything" by Joel Spolsky, *Joel on Software* – https://www.joelonsoftware.com/2005/12/13/how-to-ship-anything/

1375. "Seven steps to remarkable customer service" by Joel Spolsky, *Joel on Software* – https://www.joelonsoftware.com/2007/02/19/seven-steps-to-remarkable-customer-service/

1376. "My Style of Servant Leadership" by Joel Spolsky, *Inc.* – https://www.inc.com/magazine/20081201/how-hard-could-it-be-my-style-of-servant-leadership.html?partner=fogcreek

1377. "Designing for People Who Have Better Things To Do With Their Lives" by Joel Spolsky, *Joel on Software* – https://www.joelonsoftware.com/2000/04/26/designing-for-people-who-have-better-things-to-do-with-their-lives/

1378. "Designing for People Who Have Better Things To Do With Their Lives, Part Two" by Joel Spolsky, *Joel on Software* – https://www.joelonsoftware.com/2000/04/27/designing-for-people-who-have-better-things-to-do-with-their-lives-part-two/

1379. "Designing for People Who Have Better Things To Do With Their Lives, Part Three" by Joel Spolsky, *Joel on Software* – https://www.joelonsoftware.com/2000/05/08/designing-for-people-who-have-better-things-to-do-with-their-lives-part-three/

1380. *Microsoft Copilot: Your AI companion* – https://copilot.microsoft.com/

1381. "The Guerrilla Guide to Interviewing" by Joel Spolsky, *Joel on Software* – https://www.joelonsoftware.com/2000/03/23/the-guerrilla-guide-to-interviewing/

1382. *STEVEN ALMOND* – https://www.stevenalmond.com/

1383. *Amazon* – https://www.amazon.com/Not-That-You-Asked-Obsessions/dp/0812977599/ref=sr_1_2?ie=UTF8&s=books&qid=1234733932&sr=1-2

1384. "Tattoo Cartoon #4704", *Andertoons* – https://andertoons.com/tattoo/cartoon/4704/id-like-drunken-college-kid-right-here

1385. "Pick one and own it" by Jason Cohen, *A Smart Bear* – https://longform.asmartbear.com/one-benefit/

1386. "Mystery of a broken A/B test", *YouTube* – https://www.youtube.com/watch?v=FaUO2-AQmr0&feature=youtu.be

1387. "Hammy the Hamster Goes Organic", *YouTube* – https://www.youtube.com/watch?v=8z8CWdRaQpw&feature=youtu.be

1388. https://injuryfacts.nsc.org/all-injuries/preventable-death-overview/odds-of-dying/

1389. "Exercise Cartoon #8888", *Andertoons* – https://andertoons.com/exercise/cartoon/8888/hamster-exercise-wheel-class

1390. "Null hypothesis", *Wikipedia* – https://en.wikipedia.org/wiki/Null_hypothesis

1391. https://en.wikipedia.org/wiki/Pearson's_chi-square_test

1392. "1.3.6.7.4. Critical Values of the Chi-Square Distribution" – https://www.itl.nist.gov/div898/handbook/eda/section3/eda3674.htm

1393. https://commons.wikimedia.org/wiki/File:Game_Neverending_2008.png

1394. "SmartBear Software - The confidence behind your code.", *SmartBear Software* – https://smartbear.com/

1395. https://www.ericsink.com/articles/Product_Parenting.html

1396. https://www.archive.org/web/web.php

1397. "CVS and Perforce diffs, code review, and analysis" by Tools for CVS and Perforce – https://web.archive.org/web/20030324094126/http://codehistorian.com/

1398. "SourceSafe, Perforce, CVS: diffs, code review, and analysis" – https://web.archive.org/web/20040226150835/http://codehistorian.com/

1399. "Code Review, Source Code Metrics and Reports" by Smart Bear, LLC – https://web.archive.org/web/20041202024632/http://www.codehistorian.com/

1400. "Smart Bear Software" by Smart Bear Inc – https://web.archive.org/web/20060102014905/http://codehistorian.com/

1401. "Smart Bear®: Leaders in code review and SCM integration" by Smart Bear Inc – https://web.archive.org/web/20071010021950/http://www.smartbearsoftware.com/

1402. "Smart Bear®: Peer Code Review" – https://web.archive.org/web/20081013093635/http://www.smartbearsoftware.com:80/

1403. https://blog.businessofsoftware.org/2008/11/bos-digest-on-product-management.html

1404. "Reversi", *Wikipedia* – https://en.wikipedia.org/wiki/Reversi

1405. "File:Othello Spielbrett.jpg - Wikimedia Commons", *Wikipedia* – https://commons.wikimedia.org/wiki/File:Othello_Spielbrett.jpg

1406. "Filing Cartoon #7897", *Andertoons* – https://andertoons.com/filing/cartoon/7897/trays-on-desk-actions-and-equal-opposite-reactions

1407. "Is the Opposite of Your Choice Stupid on its Face?" by Roger Martin, *Medium* – https://rogermartin.medium.com/is-the-opposite-of-your-choice-stupid-on-its-face-5b247ffd7f94

1408. "Collaborator - Code Review Tool for Teams", *Login* – https://smartbear.com/product/collaborator/

1409. https://www.codinghorror.com/blog/archives/001058.html

1410. "Sole proprietorship", *Wikipedia* – https://en.wikipedia.org/wiki/Sole_proprietorship

1411. "Limited liability company", *Wikipedia* – https://en.wikipedia.org/wiki/Limited_liability_company

1412. https://codehistorian.com/

1413. https://www.smartbearsoftware.com/codereviewer.php

1414. https://www.midasnetworks.com/