

The three kinds of leverage that anchor effective strategies

Strength · Differentiation · Durability

“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 strategy² requires unassailable, hyper-differentiated, completely unique moats.³ 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.



“I dunno, what if we leveraged something.”

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

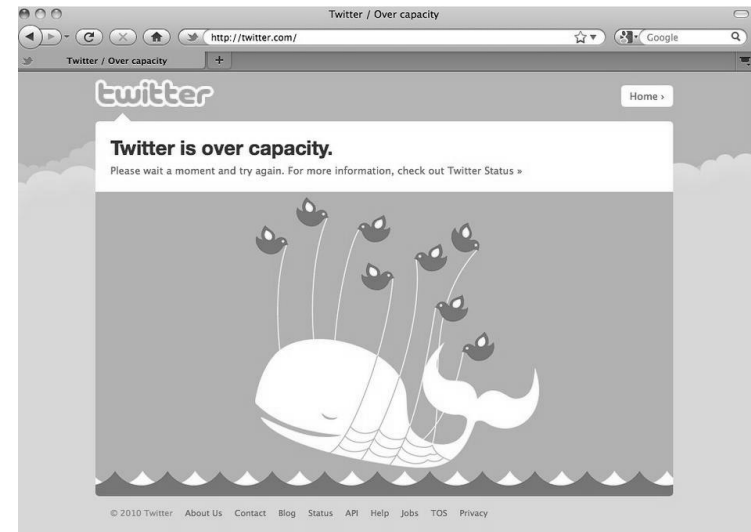


Figure 1: Twitter Fail Whale circa 2010

famously made this point when they pioneered the catchy motto “Move fast and break things,” explaining that rapid innovation is more important than anything else (Figure 2).

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.²



Figure 2

This is why typically the best answer to “what tool/language/framework/process should I use to do _____?” is “the one you already 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.

* 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.

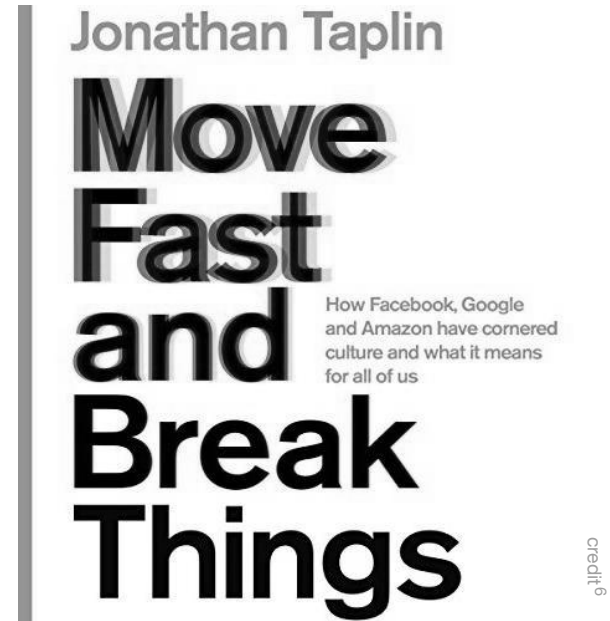


Figure 3

Better is to avoid weaknesses through decisions on target market, target personas, and product.⁸ So if your weakness is that you 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,¹⁰ * while written about infrastructure, it’s

* 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 over-



Figure 4: No one wrote a book about the new motto. Even Zuck couldn't bring himself to write out the word "infrastructure."

a general-purpose system for managing things that should be satisfied rather than maximized.¹¹

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

investing 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 emotions running 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.



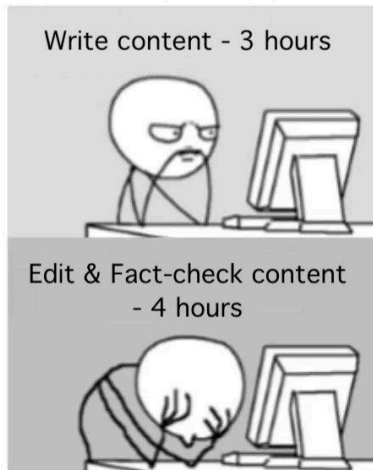
"Flight, super strength, super speed, bullet proof, heat vision, freeze breath... And you're upset because you can't see through lead?"

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).

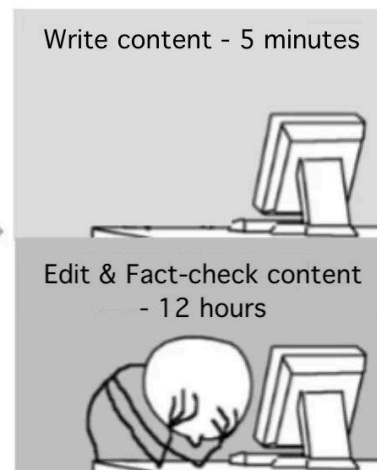
Leverage can also come from supporting tools. As of this writing, AI can generate a simple blog post, albeit with factual errors, no personal examples, and a plodding style. But, AI can *draft* ten articles, and a human can edit those many times faster than generating them from scratch, resulting in 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

Days before OpenAI



Days after OpenAI

credit¹²

drudgery of working through challenges. This is why “follow your passion” is correct but incomplete.

“*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 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 moat³ (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.

* 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),¹³ 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.

** 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.

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 prevents 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,”¹⁴ 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.

* “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 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.

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

The word for a durable differentiated advantage is a “moat.”³ 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 moats³ 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.¹⁵

- 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 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)
 - Quoting¹⁷ 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

* 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.¹⁶

** Except Apple of course; still, 87% of global handsets are Android, and Apple arguably remained because of the hardware more than the operating system.

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 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).

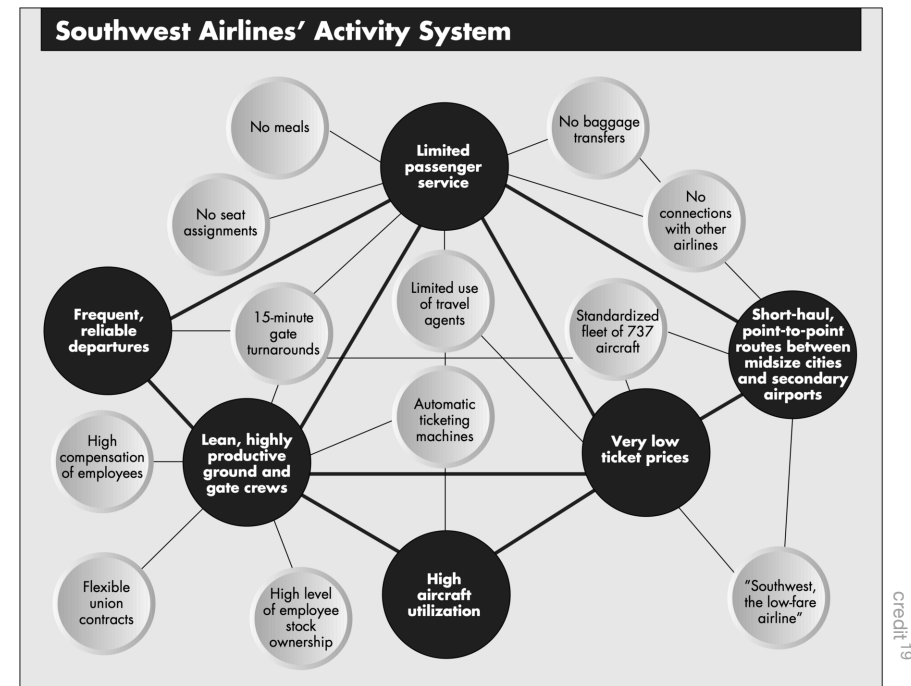


Figure 5: Michael Porter's "Activity Systems Map" for Southwest Airlines

“Strategy is a set of interrelated and powerful choices that positions the organization to win.”

—Roger Martin²⁰

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

* This idea is also explored in Blue Ocean Strategy,¹⁸ though it originated years earlier as in the example below.

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 strategy² 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.

Current version of this article:

<https://longform.asmartbear.com/leverage/>

More articles & socials:

<https://asmartbear.com>

© 2023 A Smart Bear Press

REFERENCES

1. <https://andertoons.com/sales/cartoon/8465/dunno-what-if-we-leveraged-something>
2. <https://longform.asmartbear.com/great-strategy/>
3. <https://longform.asmartbear.com/moats/>
4. <https://www.flickr.com/photos/remkovandokkum/5151041232>
5. https://commons.wikimedia.org/wiki/File:Mark_Zuckerberg_-_Move_Fast_and_Break_Things.jpg
6. <https://www.audible.com/pd/Move-Fast-and-Break-Things-Audiobook/B071NFMZZQ>
7. https://commons.wikimedia.org/wiki/File:Mark_Zuckerberg_-_Move_Fast_with_Stable_Infra.jpg
8. <https://longform.asmartbear.com/problem/>
9. <https://andertoons.com/superman/cartoon/8360/superman-psychiatry-upset-cant-see-through-lead>
10. <https://sre.google/workbook/implementing-slos/>
11. <https://longform.asmartbear.com/maximizing/>
12. <https://www.embarque.io/>
13. https://db-engines.com/en/ranking_trend/system/MySQL%3BPostgreSQL
14. <https://genius.com/Peter-thiel-zero-to-one-chapter-8-secrets-annotated>
15. https://en.wikipedia.org/wiki/The_Innovator%27s_Dilemma
16. <https://longform.asmartbear.com/willingness-to-pay/>
17. <https://www.ft.com/content/2d48d982-80b2-49f3-8a83-f5afef98e8eb>
18. <https://www.blueoceanstrategy.com/what-is-blue-ocean-strategy/>
19. <https://hbr.org/1996/11/what-is-strategy>
20. <https://rogerlmartin.com/thought-pillars/strategy>