

## Explore vs Execute

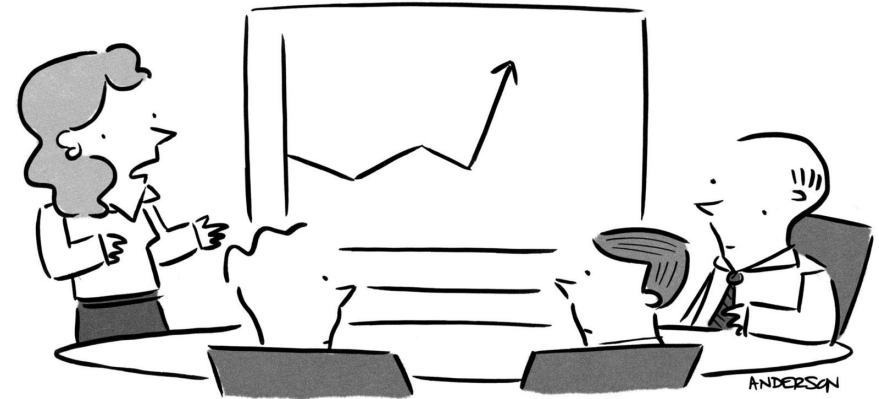
What got us here · Explore vs Execute · Cultural transition

### 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<sup>2</sup> before everyone finds out they’re a fraud<sup>3</sup> and they crack.<sup>4</sup> 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.<sup>5</sup> It was such a hard scramble to get here, against all odds, and now it’s actually *working!* Sure there was luck,<sup>6</sup> but mostly there were many good decisions, and a great



“Things are looking good, but we need to be careful to avoid any feelings of pride. Remember what happened last time?”

credit 1

product that solves a real problem,<sup>7</sup> 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 almost every thing at the company must change and adapt. 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 instead 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, 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

\* 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.<sup>8</sup>

\*\* In January 2012 at WP Engine, when we hit Product/Market Fit as documented in a separate article,<sup>9</sup> 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.

people and processes and workflows. Many of these facets are detailed in this article.<sup>10</sup>

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,<sup>11</sup> Clayton Christensen,<sup>12</sup> and James G. March.<sup>13</sup> 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,<sup>14</sup> 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 things are critical for us to win” 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<sup>15</sup>—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
short-term tactical planning	→	long-term strategic planning
new	→	valuable

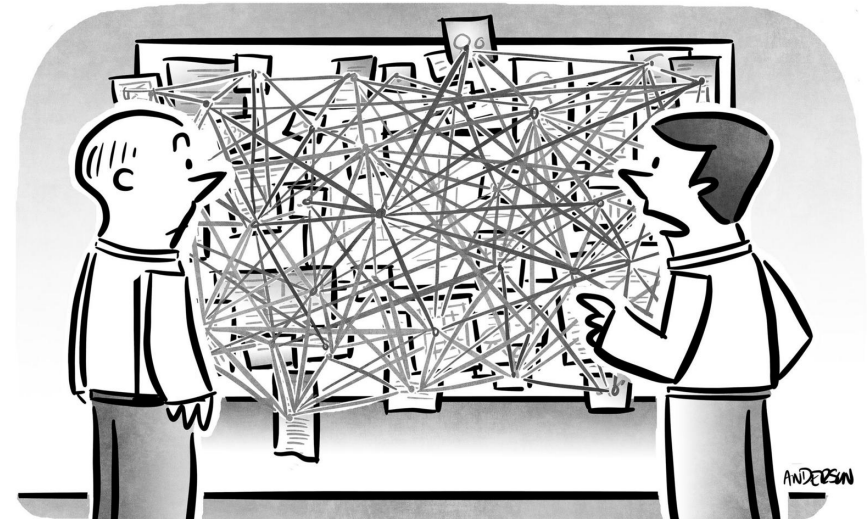
Before Product/Market Fit, you're figuring out what works. That means answering questions like:

- Who is the perfect customer?<sup>16</sup>
- Why aren't all of them buying?<sup>17</sup>
- What is the best niche?<sup>7</sup>
- Which few features make them buy?<sup>18</sup>

- What makes them want to buy?<sup>19</sup>
- What is delightful?<sup>20</sup>
- What distribution channel will work?<sup>21</sup>
- What makes us different?<sup>22</sup>
- What does the customer value?<sup>23</sup>
- What is the customer trying to accomplish?<sup>24</sup>
- What is the right price?<sup>25</sup>
- What are our biggest risks?<sup>26</sup>
- How do we avoid the obvious fatal mistakes?<sup>15</sup>

The "Explore" modality is appropriate for answering these questions.

Trying, testing, failing quickly<sup>28</sup> when it's wrong, which it often is. Moving quickly, because we're running out of time and money. Undoing errors.<sup>29</sup> Looking for big shifts rather than subtle changes that are too small to be noticed,<sup>30</sup> especially since we don't have enough data to



"At first I was trying to solve a crime. Now I think I'm making an afghan."

actually measure those changes. Being confident about nothing;<sup>31</sup> questioning everything; seeking the difficult truths.<sup>32</sup>

Exploration does not follow a timeline nor takes a singular path, although leveraging well-researched patterns<sup>9</sup> 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**

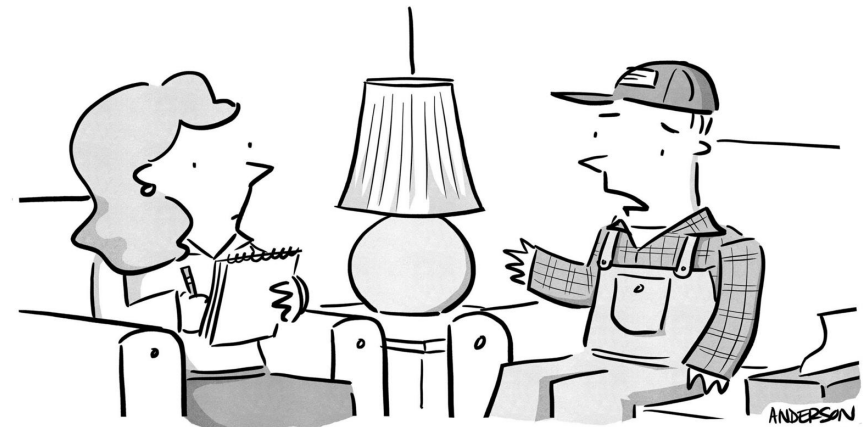
Executive roles transition from building things themselves to building teams<sup>33</sup> that build those things. Everything important enough to be done, must be done by a team, not by a single person. One reason is the quantity of work; another is for stability, so that someone can get sick or leave 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,<sup>34</sup> prioritization,<sup>35</sup> coordination,<sup>36</sup> and balancing the needs of the organization<sup>37</sup> 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,<sup>38</sup> and incurred tech debt<sup>39</sup> 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<sup>26</sup> because it hadn't figured out a business model that works, but now the thing that will cause failure is operations. Tech support<sup>41</sup> becomes bad, the product is too buggy, the product falters under load,<sup>42</sup> you can't keep



“Sometimes I put all the eggs in the same basket just to feel alive.”

up with marketing,<sup>21</sup> 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 veil 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.<sup>42</sup> 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<sup>43</sup> 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.

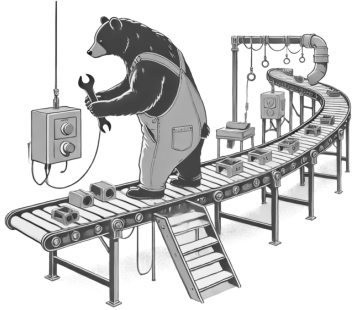
### Measuring "success"

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 advantages 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<sup>44</sup> of "Execute".

---

\* 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.

\*\* 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.



### Different people, different skillsets

Consider the personality of the person<sup>45</sup> 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.

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.

---

*Current version of this article:*

<https://longform.asmartbear.com/explore-execute/>

*More articles & socials:*

<https://asmartbear.com>

© 2025 A Smart Bear Press

---

## REFERENCES

1. <https://andertoons.com/sales/cartoon/6927/things-good-but-avoid-pride-remember-what-happened-last-time>
2. <https://longform.asmartbear.com/chaos-at-start/>
3. <https://longform.asmartbear.com/impostor-syndrome/>
4. <https://longform.asmartbear.com/startups-emotionally-draining/>
5. <https://longform.asmartbear.com/product-market-fit/>
6. <https://longform.asmartbear.com/lucky/>
7. <https://longform.asmartbear.com/problem/>
8. <https://longform.asmartbear.com/startup-beats-incumbent/>
9. <https://longform.asmartbear.com/product-market-fit-formula/>
10. <https://longform.asmartbear.com/scale/>
11. <https://rogermartin.medium.com/balancing-exploration-and-exploitation-adb82146837?ref=anand-panicker.com>
12. <https://www.hypeinnovation.com/blog/8-principles-of-the-innovators-solution>
13. <https://www.jstor.org/stable/2634940>
14. <https://www.linkedin.com/pulse/deep-dark-thoughts-being-different-dharmesh-shah/>
15. <https://longform.asmartbear.com/avoid-blundering/>
16. <https://longform.asmartbear.com/icp-ideal-customer-persona/>
17. <https://longform.asmartbear.com/put-down-the-compiler/>
18. <https://longform.asmartbear.com/the-important-thing/>
19. <https://longform.asmartbear.com/willingness-to-pay/>

20. <https://longform.asmartbear.com/creativity-over-optimization/>
21. <https://longform.asmartbear.com/first-marketing-channel/>
22. <https://longform.asmartbear.com/one-benefit/>
23. <https://longform.asmartbear.com/more-value-not-save-money/>
24. <https://longform.asmartbear.com/needs-stack/>
25. <https://longform.asmartbear.com/pricing-determines-your-business-model/>
26. <https://longform.asmartbear.com/startup-drake-equation/>
27. <https://andertoons.com/detective/cartoon/8509/at-first-solve-crime-now-making-afghan>
28. <https://longform.asmartbear.com/fail/>
29. <https://longform.asmartbear.com/good-problems-to-have/>
30. <https://longform.asmartbear.com/p-hacking/>
31. <https://longform.asmartbear.com/strong-opinions-weakly-held/>
32. <https://longform.asmartbear.com/failure-to-face-the-truth/>
33. <https://longform.asmartbear.com/delegation/>
34. <https://longform.asmartbear.com/in-command/>
35. <https://longform.asmartbear.com/rocks-pebbles-sand/>
36. <https://longform.asmartbear.com/strategic-planning/>
37. <https://longform.asmartbear.com/tension-autonomy-admin/>
38. <https://longform.asmartbear.com/slc/>
39. <https://longform.asmartbear.com/software-quality-mortgage/>
40. <https://andertoons.com/egg/cartoon/7238/i-put-all-the-eggs-in-the-same-basket-just-to-feel-alive>
41. <https://longform.asmartbear.com/tech-support-is-sales/>
42. <https://longform.asmartbear.com/scale-rare/>
43. <https://longform.asmartbear.com/growth-asymptote/>
44. <https://longform.asmartbear.com/product-metrics/>
45. <https://longform.asmartbear.com/pivot-points/>