

The Fractal Guide to Growth



Our guide to growth explores the challenges and opportunities software companies and their founders face as they prepare to scale their business. In a series of essays written by former founders and senior operators, we look across product, recruiting, sales, and engineering to better understand how to succeed when it's time to accelerate towards \$100M of ARR.

fractal

For far too long, vertical SaaS companies have been unfairly characterized as businesses with limited growth prospects. By focusing only on first-order addressable markets, both founders and investors couldn't see the many opportunities for vertical software companies to dramatically expand their TAM by layering in fintech products, marketplaces, and other potent revenue drivers. They ignored the unique leverage enjoyed by vertical SaaS companies that comes from controlling the key workflow processes of their customers and the potential for these companies to own the majority of software spend for entire industries. At best, a venture-scale vertical SaaS company was the rarest of beasts; at worst, it was nothing more than a myth.

At Fractal, we've always believed that vertical SaaS companies have the potential to become large, multi-billion dollar businesses if they have the right guidance and support. We don't help launch companies in the hopes of their founders hitting a single or a double. We believe that every vertical SaaS company in our portfolio has the opportunity to hit a homerun. The runaway success of companies like Toast, Epic, Procore, ServiceTitan, Veeva, Qualia, and many others have revealed that there are untold numbers of (multi-)billion dollar vertical SaaS businesses waiting to be built.

Scaling a vertical SaaS business requires founders to successfully navigate a treacherous growth phase that is filled with unexpected challenges. Growing a company isn't simply a matter of doing more of whatever you were doing before. The success of a startup during its growth phase largely depends on whether its founders can create well-defined and repeatable—in a word, scalable—processes at every level of their organization.

That doesn't mean these processes must be rigid. As a startup scales the processes that drive the organization can—and should—adapt to the new demands placed on the company. For many founders, implementing new processes can be a painful experience. After years of hacking and improvising to land customers and ship code with a small team of early employees, even the most flexible processes may feel rigid and superfluous. But as a company starts to grow, these processes become critical to maintaining forward velocity. Without them, internal communication breaks down, product quality declines, and customers churn.

Fractal's leadership has deep experience navigating the unique challenges and opportunities that arise during the growth phase of a company that they use to inform the guidance they provide to founders in our program. In this guide, we've distilled some of their wisdom into 4 essays that explore the challenges of growing a post-Series A SaaS company, including:

- how a CTO's role changes as their engineering team grows;
- the careful balancing act that comes with hiring your first product manager;
- how to build a sales team with intention; and
- how to hire the best employees for a job you've never done.

The insights from each essay are derived from years of experience driving growth at companies like Lyft, Snapdocs, Scout RFP, Ironclad, Workday, Namely, and Indio. They contain advice that Fractal's leaders wish they would have known when they first experienced growth pains at their own companies. We hope that founders everywhere find them useful on their journey to building the next generation of massive vertical SaaS businesses.

The Three Realities of Scaling an Engineering Team

Martin Kess, Head of Engineering

Martin is a software engineering leader, currently heading up the engineering team at Fractal Software. He has an impressive track record working on a variety of projects, from warehouse logistics software for Lyft's bikes and scooters to HR and Payroll software at Namely, Pub/Sub at Google, and high frequency trading in the hedge fund space. Martin holds a computer science degree from the University of Waterloo and a master's degree from Georgia Tech.

Scaling an engineering organization is one of the most exciting—and perilous—processes in the life cycle of a startup. After years of working closely with a small team of engineers to lay the foundation for growth, the CTO finally has the opportunity to delegate some of their responsibilities to their best engineers and take a step back from the day-to-day grind of writing code. It's a chance for the CTO to turn their focus to the big picture and plot the long term technical strategy for their company. This change can feel like a breath of fresh air, but if the transition isn't carefully managed it can be disastrous for the company. In my experience, the difference between an engineering organization that scales successfully and one that doesn't often comes down to three factors: clear communication, trust, and well-defined processes.

Establishing the Right Processes

The process of scaling an engineering organization looks different for every startup, but each company will have to deal with three fundamental realities. The first is obvious: The number of engineers on the team will increase. This has important implications for both the CTO and the startup as a whole. When an engineering team consists of just a few people, it's relatively easy to keep everyone on the same page. At any given time, the CTO and engineers all know what the rest of the team is doing and communication between team members is straightforward. Yet as more team members are added it becomes increasingly challenging for the CTO to manage individual engineers and ensure that context is shared across the entire team.

Before the engineering team starts scaling, it's possible for the CTO to keep all the engineers aligned without formally established processes. The team can function fine without Jira tickets or if they were using Jira it probably wasn't necessary to be overly disciplined about the order they were addressed. Everyone on the team is expected to know what has to be done and when. Similarly, the engineering team can probably get by without robust documentation since all the engineers have been intimately involved with building the codebase from the ground up.

But once an engineering team grows larger than about 10 people it's critical to establish formal processes for how things get done. This could mean creating a document that outlines how the team works in sprints or how the team prioritizes the features they're going to ship each month. These processes don't necessarily need to be enshrined in a written document, but in my experience taking the time to explicitly define core engineering processes pays significant dividends down the road. The earlier a CTO defines these processes, the easier the scaling process will be. The important thing is for the CTO to ensure that everyone on their team has a clear understanding of the problems that they're solving so that the team's output is predictable.

Let Your Managers Hold the Reins

Establishing workflow processes isn't sufficient to sustain rapid growth on its own, however. The CTO must still contend with the fact that at a certain point the organization will simply have too many engineers for one person to manage. This brings us to the second fundamental reality of scaling an engineering team: the creation of a management strata in the engineering org chart. The exact point that a management layer becomes necessary will ultimately depend on both the needs of the startup and the capabilities of the CTO. I've known CTOs who have single handedly managed engineering teams with as many as 35 engineers, but this an exception rather than the rule. Typically, once an engineering team reaches around 10-15 people, it's a good idea for CTOs to enlist managers to help their team function smoothly.

Ideally, the new managers will be recruited from inside the startup, which has three primary benefits. First, these engineers will have more context about the organization than an external manager, which will enable them to more quickly adapt to their new role. Second, these engineers will already have buy-in from the rest of the team. Third, their promotion will send positive signals to the wider engineering organization about the potential for career advancement within the company.

But CTOs must also be careful when promoting engineers internally for management roles. The major risk is that they accidentally turn their best engineers into mediocre managers, which will create a drag on the company's engineering output. To mitigate this risk, the CTO should structure the new position so that it is more of a parallel career path rather than a promotion to a completely new role. The new managers should still be intimately involved with the production of code, just with added responsibilities around mentoring and managing junior engineers.

One of the most common challenges that CTOs face when creating a management team for the first time is accepting the fact that this will frequently result in suboptimal results in the near term. As a cofounder of the company, the CTO has far more context about the way the business functions than the rest of the engineering team. The CTO has spent countless hours talking to customers and advisors, and as a result has a deep knowledge of both their business and their industry. But even if the CTO promotes the most talented engineers to a management role, they're still going to make worse decisions on average than if the CTO were to make those decisions on their own.

The new managers may take several additional steps to get to the same or slightly worse result than the CTO, which can tempt the CTO to step in and do the task or make the decision themselves. But this intervention can be a costly mistake.

When the CTO overrides the decisions of their newly appointed managers they never learn how to effectively manage and make important decisions. Instead, they foster a learned helplessness in their managers who are being taught to defer decisions to the CTO rather than making them on their own. This creates a lot of challenges as the engineering team continues to scale because the CTO will increasingly rely on their managers to make important decisions. But rather than making those decisions the managers will simply wait for someone else to make them. To overcome this risk, the CTO must accept the fact that their managers will make suboptimal decisions as they adjust to their new role. This is an unavoidable part of the learning process and the CTO must trust that these engineers are smart people whose decisions will be directionally correct and will gradually improve as they gain more experience.

How to Replace Yourself as CTO

The third reality that must be faced by a scaling engineering organization is the changing role of the CTO. During the earliest stages of a company, the CTO will necessarily be heavily involved in writing code and managing individual engineers. But as new managers take over many of the CTO's day-to-day tasks, the CTO's role will increasingly focus on higher level strategy that ensures the engineering team is aligned with the overarching goals of the business. In my experience, the best CTOs straddle multiple roles—they deeply understand sales, business development, product, and so on—and this capacity becomes particularly important during the growth phase of a company. If a CTO successfully scales their engineering team it can feel as though they don't really have any particular job at all. While this can feel uncomfortable at first, it is a natural evolution of the CTO role that allows them to work on higher level challenges within the company.

For example, when a startup begins to scale they will typically establish a product division that must work closely with both the engineering and sales teams. Each department will have its own priorities that must be balanced against the goals of the company as a whole. At this point, an important job for the CTO is to facilitate cross-department interactions so that the engineering team can enable their success. For instance, if the product team may not understand the engineering complexity involved with bringing an ostensibly simple feature into production. In this case, it is important for the CTO to establish the true cost of adding that feature and offer potential alternative solutions that can meet the product team's needs within the constraints of the company's limited engineering resources.

For the CTO, clear communication both within the engineering team and between the engineering team and other departments is the key to success. And one of the best ways to ensure clear communication is through ruthless prioritization. At any given time, a software company will have a long list of things it wants to accomplish. An extremely common failure mode for startups is the failure to prioritize these needs. This results in the startup attempting to accomplish all of them at once by simply hiring more engineers. While the logic behind this approach makes sense—more engineers should in principle be able to accomplish more tasks—it rarely works this way in practice.

Instead, the CTO must work closely with other department leaders to create clear priorities for the engineering team. Once these priorities are established, the CTO is better equipped to determine how those jobs should be done and who should do them. They can then communicate this information to their engineering team in such a way that everyone understands what they need to be doing. Setting clear expectations for each engineer and the engineering team as a whole is absolutely essential to the long term success of the company. Every engineer should know what their individual and team responsibilities are, and have a clear understanding of what success and failure looks like. If this information is not communicated clearly, it will result in a dramatic decline in engineering output, which startups often try to fix by throwing more bodies at the problem. But unless these communication challenges are overcome, adding more engineers who don't know their responsibilities and goals will typically only make the problem worse. Each additional engineer only adds more complexity and confusion to the mix.

This runs counter to the beliefs of many CTOs who equate successful scaling with larger teams. There is a persistent belief that a company with 100 engineers is twice as effective as a company with 50 engineers. This couldn't be further from the truth. Startups can accomplish amazing feats of engineering with relatively small teams and growing teams without the proper processes and organizational structures in place can be a fatal mistake. Many startups realize too late that engineering output does not necessarily scale linearly. Once onboarding periods and differences in talent are taken into account, it is frequently the case that adding an additional engineer actually results in a net loss of engineering output. This underscores the importance of quality over quantity when it comes to hiring engineers and establishing effective processes before scaling.

So how can a CTO know if they are scaling their engineering organization sustainably? A good test is whether they can take a weeklong vacation. If the engineering team's output continues apace, this indicates that each engineer has a good understanding of their individual responsibilities and that the overarching goals of the startup have been clearly communicated to their managers. Achieving this level of self-sufficiency is not an easy task. The important thing is for CTOs to continuously attempt to replace themselves by asking how the critical tasks they're involved with can be systematized and delegated to others. This upfront effort will compound in value with each step up in the size of the engineering team and allow the CTO to take on new important challenges.

There are No Shortcuts: Intentionality and How to Scale Your Sales Org

Mike Flores, Head of Sales Strategy

Mike Flores leads sales strategy for Fractal's portfolio of vertical SaaS companies. His guidance enables founders to secure early adopters, build a repeatable sales motion, and deploy a pioneering sales team. Previously, Mike was an early sales leader at Ironclad (YC/Sequoia), Snapdocs (YC/Sequoia), and Zillow (NASDAQ: ZG). He resides in the Bay Area and constantly seeks laughs and adventure with his wife and two daughters.

Over the past decade, I've watched a pernicious idea take root in the startup world. It's an idea that suggests that scaling a software company is all about finding shortcuts. That if a founder can pull a broad list of emails and press send on an eight-step outbound email sequence, they can conjure some initial sales traction. It reduces the hard work of a high-functioning sales org into an exercise in experimenting with cheat codes, as if mashing buttons in the right order is the key to attracting and retaining customers. The truth is there are no shortcuts in sales.

If my experience with scaling B2B SaaS sales teams has taught me anything, it has taught me that the difference between a successful high growth startup and a floundering startup often comes down to a single factor: intentionality. They are intentional about the customers they pursue and how they pursue them. They are intentional about the sales reps they hire and the way they structure their team. They are intentional about the tools they use and the data they collect. Startups that successfully scale give the finger to fate and proceed according to their plan. They are intentional in everything they do.

From Early Adopters to a Scalable Sales Motion

Once a CEO has developed a perspective on their industry, validated it by closing early deals, and launched their product with these customers, they're ready to start scaling their sales organization.

This inflection point will be different for every company, but eventually the CEO will become the bottleneck in the deal pipeline. The CEO's task is now to find a few account executives that will lead the way on scaling their sales motion, starting with early adopters. These people are not easy to find. In fact, the CEO and early account executives should become experts on disqualifying opportunities. This seems counterintuitive, as the sales pipeline is hard enough to build. However, in my experience, you have to find early adopters who will persevere through the inevitable rough patches, and will put your project at the top of their priority list.

If the CEO and his first sales hires have successfully threaded the needle and secured a healthy cohort of early adopters, the exercise shifts to pattern recognition. This is by far the most important skill for enabling fast growth. A repeatable sales motion relies entirely on narrowing down an ideal customer profile that represents a broad enough selling opportunity, and then using those customers' own words to attract more similar companies. Rather than spraying lead lists and hoping something sticks, scalable sales requires the outbound team to unearth non-obvious triggers to find comparable customers. This is the art of intentional prospecting or account based sales.

Intentional prospecting is a trend we'll see gain even more traction in years to come.

At minimum, a cold outbound email has to demonstrate at least two things to the recipient:

1. why they (and only they) could have been the recipient
2. why you have the conviction that they are at (or near) an inflection point that would require adopting your solution

Doing this at scale is hard. It's why so many software sales organizations opt for volume-based sales strategies rather than putting in the effort that account-based sales requires. However, evidence from organizations I've worked with show that the win rate of an opportunity sourced using an intentional prospecting methodology is 3x higher than an opportunity sourced via an antiquated, cadence-based mass email method. Founders are duped into playing the volume game because it often seems to be working at first. Sales reps will land customers and meet quotas, but what they haven't done is build a scalable sales process. They are closing customers essentially at random, which isn't the foundation for sustainable growth. The only way a founder can tell that they have a scalable sales process is by being very intentional about the metrics they track.

Measure Everything

The trouble with playing the volume game in sales is that it doesn't result from intentionally targeting the right customers with the right product at the right time. There are no learnings here that can be used to further increase sales because account executives don't really know why they are closing prospects. Answering this question requires diligently tracking fundamental sales metrics like outbound conversion ratios and account penetration; I am constantly amazed at how few startups track even these basic sales metrics.

Without this data it is impossible for the CEO to make informed decisions about the company's sales process. This leads to a common and fatal error at many startups: rapidly scaling their sales org before they've established a repeatable and predictable sales process. Without the numbers to understand whether a sales process is working, it's easy for founders to believe that simply hiring more sales reps will result in more sales. This is a volume-based sales mindset. But if the current sales executives have low conversion rates or if there is unsustainable customer churn, this means the sales process is fundamentally broken. If the CEO hires more account executives before refining their process, this will only compound the problem and lead to failure faster.

Being intentional about the sales data that is being collected and the metrics that are being monitored prevents the startup from relying on guesswork to grow. Instead, a company culture based on intentionality and metrics allows the CEO and sales leaders to precisely diagnose and fix the problem. It also ensures that the solutions themselves are both sustainable and effective. For example, if a startup has a monthly review of conversion metrics and finds them to be too low, the temptation is to sometimes look at new tooling or methodologies to correct the course. I've certainly been guilty of this, especially when our budgets seemed infinite. But the days of growth at any cost are long gone. Today, only companies with provable sales efficiency are going to survive.

As a founder, taking a beat to come to a consensus about the root problem is an important first step. Sometimes, new tooling is indeed the solution. But more likely the solution will require more clarity around the problem you're trying to solve and how the buyer should embark on a journey with you to solve that problem. Once you understand this, it is relatively straightforward to identify ideal customers and create a scalable account-based sales motion.

With all the data and tooling that is available on the internet, there's really no excuse to not do intentional prospecting. The companies that don't put in the effort are only admitting that they don't really know who their customer is and the value they provide to them. The companies that do will understand their customers at a much deeper level and will build a scalable, high-efficiency sales machine.

Cultivating Intentionality

The challenges with scaling a sales organization don't stop once a company has proved to itself that it has created a repeatable sales motion. As the company grows, its ideal customer profile will inevitably change and it must adapt its strategies to the needs of the new customer profile. At the same time, the CEO must contend with the hiccups that usually accompany passing the reins to a new sales management strata within the organization. They will be forced to make difficult decisions about how and whom they hire. Should they promote an internal AE to head of sales and risk losing one of their best reps, or should they risk taking on an outside hire who will be coming into the company blind? There are no one-size-fits-all solutions to these types of management challenges, but I can guarantee you that it will be easier to solve them if the CEO has created a culture of intentionality in terms of how AEs source prospects and track metrics.

As a startup grows, so does the complexity it must contend with. Addressing this complexity with intention becomes more challenging and founders will be tempted to avoid the near term pain of creating a plan and opt for ad hoc solutions instead. This may work for a while, but in the long term it will stunt the growth of the company and lead to even greater pains down the road.

How to Hire for a Job You've Never Done

Scott Bordenave,
Head of Portfolio Recruiting

Scott Bordenave is the Head of Portfolio Recruiting at Fractal, where he helps founders create effective and efficient processes for hiring top-tier technical talent during the earliest stages of their company. Prior to Fractal, Scott led talent acquisition teams that resulted in three acquisitions and an IPO.

Few factors are as important for determining the success of a startup as the quality of its employees. This makes mastering the recruiting process one of the most vital skills for a founder, especially as their company begins to grow beyond their first few engineers. During the earliest days of a startup, the hiring process is relatively straightforward. The jobs that need to be done to get an MVP off the ground are clear, as are the profiles of engineering candidates who will help bring the founder's vision to life. But as a company grows, its employment needs will quickly change. The founder will need to hire sales reps, product managers, marketing professionals, and many other positions to do work that they may or may not have direct experience with themselves. For founders, this creates an uncomfortable dilemma: How can they hire the best candidates for a job that they have never done?

The Perils of Outsourcing Hiring

When a startup is hiring its first engineers, founders are always intimately involved in the hiring process. They identify candidates, write the job descriptions, design the interview, and so on. The founders are the standard bearers of expectations for each hire and their direct involvement in the hiring and recruiting process ensures that each candidate meets the bar of excellence they have set for the position and their company. But as their company grows, there will inevitably be more people weighing in on the process to ensure culture fit, check candidate capabilities, and so on.

This "collective" approach to hiring has both advantages and disadvantages. The advantage is that more input on each hire will help offset a founder's blindspots or biases to ensure that the candidate does in fact check all the boxes for a given role. The disadvantage is that it will create an incentive for the founder to become less involved in the hiring process and delegate most or all of this task either to a dedicated recruiter or other managers in their company. This risk is especially acute if the company is hiring for a role where the founder doesn't have personal experience in terms of what to expect from the position or look for in a candidate, so they outsource it to someone at the company who they feel is more qualified to make these judgments.

I can't stress strongly enough that founders must resist this temptation at all costs. If they care about the future of their company, they must be meaningfully involved in the hiring process for as long as possible. Ensuring that the people at their company are the highest caliber possible is vital to their startups success, but by outsourcing the job of finding and vetting candidates to someone else they are outsourcing quality management.

This doesn't mean that the founder should continue running the recruiting process from start to finish for the entire lifetime of their company. At some point, their startup will be too large for them to handle this responsibility on their own and they will have to delegate some portion of hiring to others. But ideally they will never let someone else make the final decision on whether to hire a candidate or not, which means they will have to individually review every potential hire before signing off.

But once a company gets to 50, 100, or 1000 people, there's no way a founder can realistically review each candidate that makes it through the interview process, right? Not necessarily. Consider Google, a company whose hiring process has become infamous both for its intensity and length. In the earliest days of the search engine, Larry Page and Sergey Brin largely recruited friends and acquaintances they knew from Stanford. As Google rapidly grew, Page and Brin naturally had to outsource part of the recruiting process to other employees, but they never let go of the reins entirely. In fact, even when Google had more than 50,000 employees, Page would still personally approve each new hire during weekly meetings to review what recruiters had to say about each candidate. Page's deep involvement in Google's hiring process was a major factor in the company's success and as he told Wired back in 2011, the reason he insisted on being involved was because it helped him "know what's really going on."

How to Hire for the Unknown

Still, even if a founder commits to staying involved with the hiring process at their company, this doesn't solve their problem of figuring out how to identify the best candidates for jobs they may never have done before. How can they know if they're hiring the right people—or if they even need to hire someone for that position in the first place?

The answer boils down to a single word: clarity. Before the recruiting process even begins, the founder should have absolute clarity about the business need that will be solved by adding an additional person to their team. There is a tendency in tech to equate team size with success, which leads many companies to hire way more people than they need. This bloat destroys their runway and can often lead to demoralized employees who don't feel challenged or may not even know what they were hired to do.

The key to achieving clarity in this regard is the job description. Rather than simply copying job descriptions from similar roles at other companies, the founder should take the time to write out a detailed job description for the role. This will ensure that they understand how this hire will solve an actual need at their company as well as clarifying the nature of the problem that needs to be solved. This doesn't mean that founders should entirely ignore comparable job descriptions from other companies. These can be useful guide posts for benchmarking expectations for the position and understanding how this role can help achieve business objectives that the founder may not have considered.

Once a founder has clarity about what role they're hiring for and why, the next step is to rigorously work through the details of interviewing for that role. A common challenge faced by founders in the hiring process is being presented with exceptional candidates who are nevertheless not the right fit for the business' needs. All too often, a highly qualified candidate will blind founders to that reality and the solution is to create an interview process that ensures that the founder is getting the right signals to match the candidate to those needs.

This process will look different for each role and each startup. Typically, an interviewing process has both a behavioral and an aspirational component. The behavioral part of the interview allows the candidate to demonstrate that they have experience in the type of work they're being hired to do. These often involve questions that are structured as "tell me about a time when..." The aspirational part of the interview is about understanding how a candidate would react to a particular situation or approach a challenge that is relevant to the company. Depending on the role, the interview process might lean more heavily on the aspirational or behavioral components, but ideally each interview should have a mix of both. By having clarity about the business needs the founder is hiring for, they can ensure that the interview questions elicit meaningful signals about the candidate's potential to meet those needs.

Hire Fast, Hire Intentionally

Every startup has a window of opportunity that they must seize in order to sustainably grow. This makes it critical to hire the right people for the right jobs at the right time.

While rapidly growing a team is not a problem on its own, a founder must make sure that they are hiring with intention and not hiring more people simply because they have the money to do so. Clarity around business needs and hiring processes will mitigate the risk that a startup needlessly grows too fast while guaranteeing that the company has enough people to build the founders' vision.

Recruiting the first employees for a new startup is particularly challenging because each candidate is essentially taking a leap of faith on the founders themselves. The company doesn't have a track record of success yet and may not even have any employees that a candidate can use to gauge the culture and odds of a good outcome. But once a founder has a few employees at the company and MVP out in the world, recruiting more people to the team becomes easier. For one thing, they can tap into their employees' network to find high quality candidates that their employees already have experience working with. The experience of working with someone for a single day can often tell you more about a candidate than endless rounds of interviews. Moreover, candidates tend to have a perception that there is safety in numbers—if other people have joined the company and the company seems to be doing well, it seems like less of a risk for them to join as well.

These advantages enable a dramatic increase in recruiting momentum that can take a startup to the next level. The important thing is for founders to have absolute clarity about who they're hiring and why they're hiring them. A great product won't mean much if a founder can't find the right people to build and sell it. The best way for founders to ensure this happens is by staying involved in the hiring process and insisting on excellence from all their candidates.

What We Talk About When We Talk About Product

Chris Crane, Head of Product

Chris is a mechanical engineer by training and began his career in manufacturing at ERICO working to redesign products, machines, and processes for efficiency. After a wonderful stint as a stay-at-home dad, he co-founded ScoutRFP where he led the product and security teams. Following ScoutRFP's acquisition by Workday, he joined Fractal as Head of Product to get back to the basics of helping smart people create intuitive SaaS products and improve the lives of their customers.

The product team is one of the most versatile organizations within any growing startup. These are the people who are responsible for deeply understanding customer needs, translating those needs into new products, and ensuring that customers successfully use those products once they launch. The product team spans the gap between the sales org and engineering teams to drive new sales and retain existing customers. A well-run product team is absolutely essential to a startup's long term success, but many founders struggle to build effective product teams as they start to grow.

It's no mystery why. In the earliest days of a startup, when the company might just consist of the cofounders and a handful of engineers, one of the cofounders is the product team. This is usually the CEO, but many of these functions may also be handled by the CTO if they happen to be product oriented. For a seed-stage startup, it doesn't matter which cofounder handles the company's product vision. The important thing is that at least one of the cofounders owns this area and resists passing off this crucial responsibility to a "hired gun." At this point, cultivating a deep understanding of your customer is simply too important to outsource to someone else.

But as the company finds its product-market fit, things start to change. Once a startup has attracted a few customers and is starting to grow their engineering team, it becomes increasingly challenging for a founder to manage all the product duties on their own. The fact that they've sold to several customers suggests they understand those customers' needs and have at least a rough idea of their product's trajectory. If the company is going to keep growing, however, the founder in charge of product decisions is going to have to delegate some of those responsibilities. This transition is delicate and requires strategic planning on the part of the founder.

Hiring Your First Product Manager

Before we get into what founders should look for in their first product manager, we should briefly discuss some things to avoid. First and foremost, founders should resist the temptation to pass ownership of product decisions to their engineers. While product management is certainly a teachable skill, this naturally tends to result in engineers making product decisions from an engineering perspective. Engineering-led product decisions may be technically elegant and may even meet some customer needs, but it's not the right approach when a startup is focused on growth. Instead, founders should seek product managers with a strong background in business or design and—ideally—your target market.

Another common mistake founders make when hiring their first PM is they hire someone who is too senior too soon. At first, this seems counterintuitive. A senior PM will come with a lot of experience that can be very valuable for a startup looking to rapidly accelerate their growth.

They've been there before and have a product wisdom that was battle tested in the market. The problem, however, is that hiring a senior PM early in a startup's journey often results in conflict over who owns the company's product vision and may not be willing or able to do the vast array of tactical work required of a PM at this stage. A senior PM will often expect founders to completely hand over the reins when it comes to product decisions, but this can be a risky move during the early stages of growth. At this point, no one knows the product better than the founder and they should still be involved—to a greater or lesser extent—in important product decisions. If the founder's role in the process is not clarified up front, this can lead to significant conflict. An important caveat here is that if a founder does want to take a significant step back from a lot of strategic product work, a senior PM can carry that responsibility. The founder should be aware, however, that this will typically mean providing them with executive level pay and equity.

Conversely, I think there's a lot to be said for hiring a more junior product manager who is relatively early in their career. While they may lack the experience of a more senior PM, there are some distinct advantages to this approach. First and foremost, hiring a more junior PM requires the founder to be incredibly clear about what roles and responsibilities they will be handling. If the founder still really wants to be involved with roadmapping, but wants to take a step back from user testing, they must clearly communicate this to the PM in order for them to do their job effectively. This clarity around the roles and responsibilities will also aid in the hiring process since it allows the founder to know exactly the type of experience they're looking for in the role. Generally speaking, the founder should hire someone who can offset their own weaknesses—it could be pre-sales, user experience or embedded design. Whatever it is, they can quickly narrow in on the right person to fill in those gaps.

The Role of Product During Early Growth

Once a founder has made their first product hire and started building their product team, they need to prioritize their jobs to be done. Depending on the organization, a product team may have a wide range of functions that touches on every aspect of the customer experience. But during the early stages of growth, not every product function is equally valuable. For a resource constrained startup, prioritizing the functions of their product team is paramount. And the most important role for the product team at this stage will be figuring out how to get more customers.

By the time a startup has started making their first product hires, their MVP is out in the world and they have found some semblance of product-market fit. But PMF is a slippery concept and it is not always so obvious which market the product fits. For example, I've seen startups who have successfully found traction for their products without really understanding how to turn those early wins into a repeatable sales process. The reason is because they didn't precisely understand the customer segments that were adopting their product. Without this knowledge, all they could do is take a spray-and-pray approach to sales, which is a poor strategy for accelerating growth. In these scenarios, the product team's first step in getting more customers is to do the user research that will allow them to segment the market to understand who is buying the company's product and why. As Mike pointed out in his essay, this is the foundation for a repeatable—and more importantly, scalable—sales motion.

During the early stages of growth, a startup must walk a fine line between keeping their existing customers happy and attracting new customers. While keeping existing customers happy is certainly important, if a startup over-indexes on current customers, the end result is that they will always run out of money. The key to growth is getting more customers, but simply focusing on increasing the quantity of customers isn't sufficient. Instead, as the startup attracts more customers, the value of their contracts should increase in tandem. Yet as any competent product manager knows, the needs of a customer that is paying \$10,000 per year for software is much different than the needs of a customer that is paying \$100,000 per year. It is imperative that early product hires are getting in front of larger customers to understand their needs so they can drive product decisions that deserve bigger contracts.

While the first priority of a fledgling product team is getting more customers through the door, it must also take steps to shape the customer experience to limit churn. And this requires product teams to look inward. What I mean by this is that customer success will depend to a large degree on the product team's ability to manage internal communications between sales and engineering. Contrary to popular belief, the customer experience doesn't begin the first time they login to a product. It starts with the initial demo and sales pitch, which is especially true when a startup is young and their software is likely buggy and incomplete. Managing and understanding customer expectations during the onboarding stage is crucial to keeping those customers. This requires the product team to collaborate closely with sales reps so that these reps understand how the product is evolving and can more effectively pitch customers on the future vision of the product.

At the same time, the product team should also work closely with the engineering organization to ensure that the tools they're building actually map to real customer needs. Supporting customers is necessary, but not all "support" is created equal and it is up to the product team to make that crucial distinction. When a startup is first getting its bearings in its target market, there is a real risk that valuable engineering time will be squandered on product features that don't really influence customer decisions. For example, a company I cofounded spent years manually completing various reporting tasks for customers. This obviously wouldn't scale and it was well within our engineering team's capability to build a solution. But as the leader of our product team, it was clear that this was not the most valuable use of our engineers' time at that point. We ended up building those tools eventually, but only when we had the engineering resources to do so and had already shipped the more valuable products.

An effective product team will triage support requests to ensure that only those that are most valuable in terms of revenue make their way to the engineering team, which is likely preoccupied with squashing bugs and ironing out existing software. If the product team fails here and allows all support requests to go unfiltered to the engineering team, this will lead to too much context switching, wasted time, and missed deadlines that could eventually bring down the company.

When founders are strategic about their early product hires and clear about their responsibilities, they should see a remarkably positive impact on their revenue growth. If sales aren't growing, this should be a red flag that they've missed the mark somewhere. More often than not, the answer is that they are building the wrong products for the wrong customers. But how did they get there? Do sales reps not know what features the product team is planning so that they can adapt their messaging to prospects? Are engineers busy building nice-to-have features instead of products that will land larger customers? Only by understanding the roles and responsibilities of a highly effective product organization can founders diagnose the specifics of the problem and lay the foundation for sustainable growth.

About Fractal Software

Fractal Software launches and finances vertical SaaS companies that solve real problems for businesses in overlooked industries. We unlock entrepreneurship for exceptional individuals by providing capital, a co-founder, a business idea, and ongoing support as their company grows. Together, we'll create fast-growing vertical SaaS companies that modernize the way America's small and medium businesses operate so they can better serve their communities.

Learn more at www.fractalsoftware.com