BUILD A MULTIMILLION DOLLAR STARTUP *WITHOUT* VENTURE CAPITAL

THE SAAS

PLAYBOOK

"I've been learning from Rob since the early years of HubSpot. You should too."
—DHARMESH SHAH, Co-founder, HubSpot

ROB WALLING

Co-founder of MICROCONF *and* TINYSEED

THE SAAS PLAYBOOK

BUILD A MULTIMILLION DOLLAR STARTUP *WITHOUT* VENTURE CAPITAL

ROB WALLING

Written by Rob Walling
Edited by Jessie Kwak
Cover Art by Pete Garceau
saasplaybook.com

ISBN 979-8-9877465-2-3 (ebook)

First Edition

Praise for *The SaaS Playbook*

"Rob knows software, and he knows startups. I've been learning from him since the early years of HubSpot. You should too."
– DHARMESH SHAH, Cofounder/CTO, HubSpot

"By far the best book on this subject. Packed with specific instructions but also surprisingly deep insights into the hurdles and solutions. You can tell he's speaking not only from his own extensive experience but from knowing hundreds of other startups. A must-read for every bootstrapping entrepreneur."
– DEREK SIVERS, Founder of CD Baby. Author of *Anything You Want*

"If you're starting or running a SaaS business, this book isn't just good; it's essential. Rob's seen hundreds of SaaS businesses fail and dozens succeed, and he's distilled that knowledge into a beautifully readable, perfect-sized volume of only the really good advice. Even as a seasoned founder, I kept uncovering bits of gold, chapter after chapter. You will, too. If you want to avoid the pitfalls that imperil every SaaS startup (and kill 95% of them), read this cover to cover and refer back often; that's my plan."
– RAND FISHKIN, Cofounder, SparkToro

"I do whatever Rob recommends in SaaS! With a track record like his, it's a gift he's sharing this playbook with the world. Must read for every SaaS creator."
– NOAH KAGAN, Chief Sumo, AppSumo.

"Rob's one of the few minds who sits at the center of the SaaS ecosystem

with the knowledge of not only how to build SaaS products properly but also how to scale them. His playbook is something I've wanted for years—glad he finally put it down on the page so we can all learn."
– PATRICK CAMPBELL, Cofounder, Profitwell

"Most of the advice that's out there about growing a SaaS past the initial phases is pretty bad. Not a lot of people have actually run a bootstrapped SaaS business for a long time. Rob Walling is the exception. He's been in the trenches for over a decade and has been mentoring the indie software community the whole time. I started reading this book, and I can't put it down."
– PELDI GUILIZZONI, Founder and CEO, Balsamiq

"Rob has more years in the bootstrapped trenches than anyone, and that's why he's my go-to source for how to build a lucrative bootstrapped business. After learning from his teachings and community, I exited my bootstrapped SaaS for 7 figures!"
– LAURA ROEDER, Founder, Paperbell

"Rob is a startup Grand Master who's inspired me and countless others. A chance to learn from Rob's condensed knowledge is a drop-everything-and-do-this-instead moment."
– COURTLAND ALLEN, Founder, Indie Hackers

"Rob has instilled so many important lessons about building software businesses in me so deeply that I almost forget how naive and lost I was before finding his work. Even if you wouldn't call your business SaaS, you're going to learn something from this book that changes the trajectory of your company."
– ADAM WATHAN, Founder, Tailwind Labs

"If you want to raise a million dollars for your next big idea, look elsewhere. If you want to develop your idea into a multimillion-dollar software company, this book is for you. Rob Walling is the go-to strategist for bootstrapped entrepreneurs. As a founder, he's achieved life-changing exits, all while advising and helping countless others. His ability to share his in-the-trenches experience with authenticity, clarity and intellectual honesty is unmatched."
– DAN ANDREWS, Author of *Before the Exit*

About the Author

Rob Walling is a serial entrepreneur who has started six companies (five bootstrapped, one funded). He's been teaching founders how to build, launch, and grow startups for 17 years, but he hopes that doesn't make him sound old.

He runs the largest community for non-venture track SaaS founders, MicroConf, and the first bootstrapper-friendly accelerator for SaaS, TinySeed.

His podcast *Startups for the Rest of Us* has shipped every week since 2010 and now has more than 650 episodes and 10 million downloads.

Rob has invested in more than 125 companies and has been quoted in dozens of major publications, including *The Wall Street Journal, Forbes, Entrepreneur,* and *Inc. Magazine.*

This is his fourth book about building startups. Learn more about him at RobWalling.com.

For this book's online resources or to purchase the print, ebook, or audio version, visit saasplaybook.com.

Dedication

To Sherry and the boys

Thank you for your tremendous support during the years it took me to learn what's in this book

Table of Contents

Foreword by Jason Cohen **13**

Introduction **19**

The Playbook for Building a Multimillion-Dollar SaaS **29**

- You Know What's Cool? A Million Dollars 29
- What Is Bootstrapping, Really? 31
- Why Focus on SaaS? 34
- Why Is SaaS the Best Business Model? 35
- Achieving Escape Velocity 39

Market **43**

- Strengthening Product-Market Fit 43
- How Can I Compete in a Competitive Market? 51
- How Much Should I Worry about Competition? 55
- How Can I Build a Moat? 57
- Should I Translate My Product into Other Languages? (And Other Common Mistakes) 62

Pricing **67**

- How Should I Structure My Pricing? 67
- SaaS Cheat Code: Expansion Revenue 70
- Should I Offer Freemium? 73
- Should I Ask for a Credit Card Up Front? 75
- When Should I Raise Prices? 79
- How to Raise Prices 80

Marketing 87
How Do I Find More Customers? 87
Marketing Funnels 89
SaaS Cheat Code: Dual Funnels 93
Business-to-Business SaaS Marketing Approaches 95
How Do I Know Which Marketing Approaches Fit My Business? 101
How Should I Structure Sales Demos? 109

Team 117
How Should I Structure My Team? 117
Hiring Managers 127
How Can I Hire Great People? 131
Should I Offer Equity, Stock Options, or Profit Sharing? 135
Do I Need a Cofounder? 139

80/20 SaaS Metrics 143
Which Metrics Should I Track? 143
3 High/3 Low Metrics Framework 145
SaaS Cheat Code: Virality 152
How Much Should I Worry about Churn? 154
SaaS Cheat Code: Net Negative Churn 161

Mindset 165
How Do I Achieve Success? 165
Where Should I Focus My Time? 168
Should I Raise Funding? 171
Am I Turning Speed Bumps into Roadblocks? 176
Where Can I Find Community? 179
How Can I Avoid Burnout? 182
What Are Founder Retreats? 186

Afterword 191
Acknowledgments 193
Appendix A: Resources 195

Foreword

by JASON COHEN, Founder, WP Engine

Is it OK to want to get rich with your own two hands, beholden to no one, especially not to some already rich person who wants to trade a huge chunk of the reward for a check and some advice from the cheap seats?

Is it OK to be obsessed with your work and ideas yet not want to "change the world" or "put a dent in the universe?" Is it OK if you want to make a low-chance-of-success career decision mostly because you hate being told what to do and therefore cannot, in your own words, be successful in a "real job?"

Is it OK to have the hubris of thinking your product is better than the other hundred already in-market and that it makes sense for a 101st to exist? Is it OK to spend every waking hour (including two in the morning) obsessing about your project rather than about other people, turning yourself inside out even to the detriment of health and relationships?

It doesn't matter whether it's OK, it's what you're going to do.

You've already decided to do it; that's why you're holding this book. You've already decided you have to do it, whether it's justified or not, whether for a higher purpose or a simple force of personality.

Congratulations, that makes you a founder. (Or should I say, condolences.) Join the party.

It is a peculiar party, its members bonded by experiences that can be explained but not truly understood by those who have not experienced it for themselves, like becoming a parent. Experiences like quitting your job and finding those you count as friends or family becoming doubters waiting—maybe even hoping—for you to fail.

The elation the first time a customer gives you money for something that you created, which you can hardly believe they did considering how bad the product is.

The trepidation of the first time you hire someone, where the family of a fellow human being is now dependent on you, and suddenly you realize that payroll is a permanent inexorable weight, a new stressor and a visceral demonstration that the line between "business" and "personal" is very wide and very fuzzy.

The gut-wrenching first time you fire someone, having worsened the situation by waiting far too long, having no idea how to be constructive, giving explanations that are nonsensical because they are not the true reason, knowing you are putting their entire family at risk, and afterward the others simply mutter "what took you so long?"

The constant barrage of customer complaints and compliments, of showing up randomly on Twitter and in newsletters, whether for good or ill, of judgments from anyone and everyone about how stupid it is that you did this or how genius it was that you did that, which incidentally they're wrong about half the time, although it can be hard to tell which is which.

So you possess the personality defect that drives you to join this coterie; now comes the hard work. Fortunately for you, Rob Walling has been one of the most active members of this party for a few decades, having built several successful companies and, through investment, has seen the ups and downs of more than one hundred. He distilled this experience into a book of wisdom, exactly what you need to tackle the work now in front of you.

Most of what you'll need to do, you've never done before, and you're not good at it, although you think you are. Probably you'll be better than most people at doing it for the first time because you're highly motivated and because the work is connected to other work that you do understand. So, you don't know marketing, but you do know your customer; it's better to write clearly and directly to a person you deeply understand than to have a degree in advertising.

Still, you're not good at it, whether or not you think you are a genius at everything. This delusion of competence is helpful in overcoming the otherwise overwhelming barriers to embarking on this improbable project. And in some aspects, your unacknowledged deficiencies aren't detrimental—you don't need to be an expert in E&O insurance when no customers are depending on your software, nor in accounting when there's no revenue to account for.

But there are key areas, early in the life of a bootstrapped company, where lack of competence is a leading cause of death. Dozens of twittering successful bootstrappers agree, for example, that picking the right market was the primary cause of their success; many self-reported autopsies blame the converse. Pricing is another example; in my case, the moment when hyper-growth ignited in my most successful company was exactly coincident with a customer-inspired revamp of our pricing and packaging.

Focusing your attention and your learning on these specific areas that are covered in this book—market, pricing, and also marketing/sales, team, metrics, and personal mindset—gives you the best chance to succeed. Success is still not guaranteed, but these are the areas where it's most vital to make good decisions. You can make the best possible decisions in poker and still lose; that's the nature of systems—like poker and startups—that are equal parts skill and luck. In such systems, you must be diligent and intentional in making the best possible decisions, recognizing that luck is a real but uncontrollable force. And you need the wisdom that can only come from seeing the movie play out hundreds of times in different ways, which is exactly what Rob has done, not only through direct experience but as a writer, thinker, interviewer, and synthesizer. There is no better way to maximize your probability of winning.

You chose this path because you don't like to be told what to do, and here's a book telling you what to do. But it's not forcing you to do things. It's giving you Cheat Codes in the areas that are crucial to get right and in which—for the good of the company—you have to admit you are not a world expert.

And if you discover any of it wasn't right for you, you can wave your thriving business in Rob's face and tell him how wrong he was. Won't that be fun?

So read this book. Use it to become a better version of the person you already are, and give you and your business the best chance at success.

JASON COHEN, Founder, WP Engine

Introduction

My phone buzzed. It was an incoming email.

I never have email notifications turned on, except for two weeks in late June of 2016. The startup I had launched two and a half years prior had grown into one of the top 10 companies in our space and was in the final throes of being acquired.

Some acquisitions are scary. All of them are stressful.

This one was going to be life-changing, which made it even worse. If it didn't go through, I'd be back in the startup grind. Restless sleep. Loads of stress. Financial uncertainty.

For something I had dreamed about for decades, running a fast-growing bootstrapped startup was more of a mixed experience than I'd anticipated. There was absolutely fun to be had ("These are the good old days," I used to remind the team), but the struggle was real. And it was taking a toll on me.

My mind flashed to the day we crossed $1 million in annual revenue. The day we hired our eighth employee. The day we finally moved into a real office with huge windows overlooking a busy downtown street. In each of these moments, I felt like I'd made it.

I then flashed to my "lost" six months in 2014 when I thought of nothing else except how I was going to make payroll. The moment I broke down sobbing in my parked car because of the stress of the acquisition. Or the moments I yelled at my wife or kids for things that were far more my fault than theirs.

And yet there I sat, in a college classroom outside of Portland, Oregon, in late June 2016. I was at my oldest son's cello camp. He was practicing with a group of other musicians. And I had a new email notification on my phone.

"Please DocuSign: Drip Written Consent of Board
(asset sale, Drip Shareholder Consent . . .)"

"This is insanity," I thought. *"I can't believe this has come together."*

To the chagrin of the cello instructor, I motioned that I needed to step out of the room for a moment. I opened the email, signed the final documents that would be life-changing in many ways using my index finger on an iPhone screen, and clicked "confirm."

It was done. The result I had dreamed of for decades and been "all in" on for 15 years was over. I could take a deep breath and relax. At least, that's what I kept telling myself.

I opened my mobile banking app and logged into our company bank account. "That's a lot of zeros." By far the most money I had ever seen in one place. At that moment, life felt different.

Bzzzzzzz

A text message arrived from Derrick, my cofounder, with whom I'd spent the past three and a half years building Drip. He'd sent me a screenshot of the company bank balance.

Me:	Haha, I was just going to send you the same thing ;-) Wow. Congratulations, sir
Derrick:	*virtual handshake* Congrats indeed!

He followed up with a bitmoji of him in a green suit with dollar bills falling around him, which was quite out of character for my typically reserved cofounder.

I exhaled slowly. The stress wasn't going away. I hoped it would soon.

Why You Should Read This Book

There's a narrative in the startup ecosystem that assumes you will raise funding when you start a company.

I've attended meetups where I mentioned bootstrapping my company to millions in revenue and was asked, "Why would you do that?" From the near-mythical origin stories of Apple, Facebook, and Google to the wildly popular TV show *Shark Tank*, funding is the assumption rather than the exception.

In the introduction of my first book, written in 2009, I said the following:

> “ *I am not anti-venture capital. I am anti-everyone-thinking-venture-capital-is-the-only-way-to-start-a-tech-company.* ”

The dream of being picked from a sea of wannapreneurs, anointed as a "real" founder, and handed buckets of money is alive and well in Silicon Valley and other startup hubs around the world. Except there's something wrong with seeking this narrative . . . It's lazy.

It implies that you need someone else's permission to build your company. That you're not a real entrepreneur until an investor tells you that you are.

Or maybe you like having an excuse not to ship, and a never-ending quest for funding is a pretty good excuse. There's a reason the most common piece of advice I give aspiring founders is:

Build your business, not your slide deck.

Instead of waiting for a basket of money to fall into your lap, go build your business. If you were an author, I would tell you to stop asking publishers for permission and go write your book. Andy Weir (author of *The Martian*) didn't wait for approval; he wrote an international bestseller that's been made into a film starring Matt Damon.

If you were a filmmaker, I would tell you to stop asking film studios for permission and go make a film. Kevin Smith and Robert Rodriguez did, and they've built careers based on their unique voices and scrappy approaches to filmmaking.

Much like the author who waits to write their book, or the filmmaker who waits for permission from the studio, the startup founder who waits for funding to start their company is more likely to wind up disappointed than funded. At least, that's the way the numbers play out.

If you search for meetups about startups, most will assume you are seeking funding. If you search the Internet for how to launch a startup, the first step is usually building a slide deck to pitch to investors. Instead, I would tell you to focus on building your business.

The point of this book is to show you that path. If raising funding is the blue pill, I invite you to take the red one. It may not be pleasant, but it's a reality you control. A reality where you don't need permission.

After nearly two decades of working with startup founders, I know that bootstrapping takes longer to generate life-changing wealth than the moonshot approach of raising venture capital. But the likelihood of some kind of a "base hit" is much, much higher. You might think of it as a .01% chance of making $100 million versus a 20% or 30% chance of making hundreds of thousands or millions (and maybe a higher percentage if you don't make the common mistakes).

More than 99% of companies that seek funding do not receive it, and the vast majority of those that land funding ultimately fail. This is not the case with bootstrapping, as I've observed first-hand while starting six startups and running the largest community for bootstrapped (and mostly bootstrapped) software founders, MicroConf.

There are superpowers to being bootstrapped. One is that you don't need anyone's permission to start or build your company. Another is that your business doesn't die until you quit. Bootstrappers don't run out of money; they run out of motivation.

It's a shame that although more than 99% of new companies started every year take no funding, every article you read about starting a tech startup assumes and glorifies the raising of a huge funding round. As if raising funding is the goal—it's not.

Building a business that generates enormous profit and serves its customers, founders, and employees should be the goal. As a founder, you're likely seeking freedom from working for others or wealth to support your lifestyle.

Maybe you want to spend 10 hours a week working on your company and the rest of the time traveling or with your family.

Or maybe you want to earn millions of dollars so you can buy your dream car, your dream house, or Banksy NFT.

Maybe you just want a sane work life in which you have more say about when and how much you work.

There are many reasons to start your own company, and funding is simply a tool you may or may not opt to use. The good news is the easiest way to raise funding is to build a great business first.

I wish the tech press spent more time selling us on this dream and less time telling us how a company that will be out of business in 18 months just closed its Series A (at a ridiculous valuation, no less; otherwise it wouldn't be on the front page of TechCrunch). Aspiring founders read this and figure that if they play the startup lottery, one day they, too, can be anointed as worthy.

My intent is not to dissuade you from raising funding (I run a fund that invests in startups, for crying out loud). My hope is to expose you to the massive, unseen part of the startup ecosystem that exists beyond the tech press. The part that houses the more than 99% of startups that don't ask for permission to start.

Instead, they show up, they work hard, they focus, and they ship products every day. They made the decision to build their business instead of their slide deck. This is what I did on my 13-year journey bootstrapping software products and software as a service (SaaS) companies. Tens of thousands of others are doing the same today.

Many will become successful, and some will become wildly profitable. They won't get there by waiting around or asking permission. They show up every day, and they do the work.

Who Should Read This Book

If you want to build an ambitious software startup with or without raising capital, you should read this book.

More specifically, I will focus on SaaS. Think of SaaS companies as real products used by real customers who pay them real money (usually monthly or annually). You're likely aware of many SaaS companies: MailChimp, Basecamp, Dropbox, Slack, and Zoom, to name a few.

If you're looking to grow a side project that could one day replace your full-time income (and then some), you should read this book.

If you're a software developer, a no-code aficionado, or you've never written code in your life, but you're looking to learn the most actionable tactics and frameworks to build and launch a SaaS, you should read this book.

If you're looking to learn how to build a product people want and are willing to pay for or how to reach customers at enough scale to grow your company into a seven- or eight-figure business, you should read this book.

Who Shouldn't Read This Book

If you are dead set on raising venture capital and building a high-risk billion-dollar company, this book is not for you. I won't be showing you how to hone your pitch deck, cozy up to venture capitalists, or evaluate a term sheet.

If you're looking for excuses for not making progress on your startup, this book is not for you. I won't be telling you that you deserve to be successful or that it's going to be easy. I've been doing this too long to present it any other way.

If you're looking to build a business on your own that pays your bills but doesn't grow beyond that, this book will have mixed value for you. There's a significant difference between launching a product that generates $20,000 per month and one that makes $200,000 per month, starting with foundational elements like the idea and the market.

If you want to build a business that allows you to stay small and live the four-hour workweek, I suggest checking out my first book, *Start Small, Stay Small: A Developer's Guide to Launching a Startup* (even if you're not a developer).

That book focuses on starting small software companies for lifestyle purposes like quitting your day job or minimizing the hours you need to work. You might think of the book you're currently reading as a sort of spiritual sequel to *Start Small, Stay Small.*

If you're looking for the promise of riches followed by generic advice like "build an audience and ask them what they want," "scratch your own itch," and "follow your passion," this book is not for you. It's not a regurgitation of the feel-good entrepreneurial tropes on social media.

The Playbook for Building a Multimillion-Dollar SaaS

You Know What's Cool? A Million Dollars.

I struggled with the title of this book. I'm not a fan of hard-selling, over-promising Internet marketers who use phrases like "seven-figure" or "million-dollar" because they have a nice ring to them.

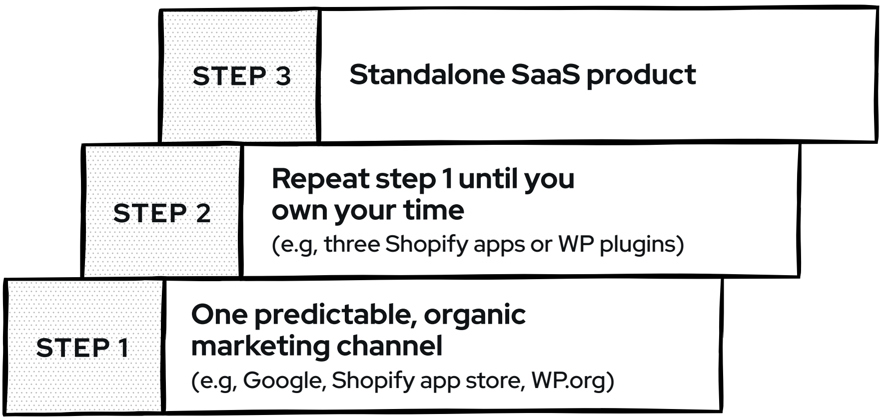
And yet, this book focuses specifically on guiding you toward launching and growing a SaaS product to seven or eight figures in revenue. So "Build a Multimillion-Dollar Startup" is the actual focus, not just marketing copy.

My first book, *Start Small, Stay Small,* focused on tiny niche products that generate thousands or low tens of thousands in revenue each month.

That's not the focus of this book.

This one is another step or two along the journey (in my Stair Step Method of Entrepreneurship, this book is focused on step 3).

The Stair Step Method of Entrepreneurship



Step 1: Your First Product
You've got a simple product with a simple marketing plan, usually with a single traffic channel, like a WordPress plugin or Shopify add-on. Basically, something that allows you to start generating revenue and learn the ropes.

Step 2: Rinse and Repeat
Step 2 involves doubling down on the model that worked in Step 1 and repeating it until you own your time. Take a look at what made that product a success and repeat it with another product. Continue getting good at the free or low-cost traffic channels that are working for you.

Step 3: Standalone SaaS Product
SaaS is complex to build and market, and there's a long on-ramp to any kind of substantial revenue. The point of the Stair Step Method is to allow you to gain experience, skills, confidence, time, and revenue before starting the difficult challenge of building a standalone SaaS product.

I developed the Stair Step Method based on patterns I saw in the bootstrapped software community, but of course, it's possible to skip the stairs and still achieve success with your product.

Whether you've bought out your time with smaller products or you want to jump into the deep end, this is the playbook for launching and growing a SaaS startup to millions in revenue without raising buckets of venture capital.

This book is filled with lessons I've learned about building companies, having done it myself, invested in more than 125 startups, and worked closely with thousands of founders through my writings, podcast, community, and SaaS accelerator.

It's a collection of the best strategies, tactics, and frameworks I know for conquering the speed bumps you'll encounter on your startup journey.

What Is Bootstrapping, Really?

As recent as a few years ago, a startup was one of two things: bootstrapped or venture-funded. Over the past few years, more options have become available, even though most people still think in terms of the bootstrapping versus funding dichotomy.

Here are terms the startup community uses to describe the funding status of a company:

Bootstrapped: You started the business on your own, with limited resources, and grew it slowly over time as the business generated cash. This is the longer road, but it allows you to maintain complete control of the business and not answer to investors.

Venture-funded: You found investors (sometimes friends and family, but usually angel investors or venture capital firms) who were willing to write you a check to grow your company and raise your next round of funding in about 18 months. Rinse, repeat, and sell the company or IPO seven to ten years later for at least a billion dollars.

Self-funded: Some use this interchangeably with bootstrapping. I define it as being able to fund your next startup with your current resources.

Maybe you have a product generating $30,000 a month, so you're able to invest $15,000 a month into your next effort. I did this when I started an email service provider called Drip. I invested between $150,000 and $200,000 of the profits from a prior SaaS company into building, launching, and growing Drip before we reached profitability.

I had that luxury because I had stair-stepped my way up from consulting during the day and building products at night to having a few small software products making $500 to $5,000 per month to a SaaS application that earned $25,000 to $30,000 per month and allowed me to enter the extremely competitive and lucrative market of email service providers.

Mostly bootstrapped: Some founders want to build a bootstrapped company but know that a bit of funding will help them get there faster. These funding rounds are usually in the $100,000 to $500,000 range, and they are raised from friends and family, non-venture-focused angel investors, and bootstrapper-friendly funds like the one I run, TinySeed.

I was blown away the first time I heard a founder talk about raising funding without the intention of following the path of traditional venture capital. That company is Customer.io, which raised its first round of $250,000 in 2012. Customer.io is now a highly profitable, eight-figure SaaS company.

They have since raised additional rounds of funding, but the founders still own the vast majority of the company, and they run it as if it were bootstrapped. It's capital efficient, maintains healthy but steady growth, and focuses on serving its customers and team members rather than its investors. I would call them a *mostly bootstrapped* company. They don't fit the technical definition of bootstrapped, but they are much closer to being bootstrapped than venture-funded.

Castos is another example. The founder, Craig Hewitt, saw a gap in the podcast hosting market and parlayed a podcast editing service into a WordPress plugin and then into a SaaS business called Castos. He grew the company profitably to six figures of annual recurring revenue before taking $120,000 in funding from TinySeed, followed a year or so later by an additional $756,000 in funding from follow-on investors.

Craig used his funding to expand their team, allowing him to grow top-line revenue and create room to expand the team even further. He didn't sign a lease on a big office in the SOMA neighborhood of San Francisco or spend $50,000 on a billboard on Highway 101. Instead, he hired slowly and grew efficiently. Thus, I call Castos a *mostly bootstrapped* startup.

So Many Terms, Which Should We Use?

In this book, "bootstrapped" includes *self-funded, bootstrapped,* and *mostly bootstrapped* startups. The difference between having no money or a bit of money is insignificant compared to the chasm between bootstrapping and raising millions in venture capital to shoot for that "$1 billion valuation or bust." It's no longer a dichotomy of funding versus no funding; it's about how you approach your company's growth.

Do you aim to build a real product that sells to real customers who pay you real money? This book calls you "bootstrapped," even if you raise a bit of money.

Do you aim to run your company near break-even, be capital efficient with your decisions, and potentially pull profits out over the long term (or sell if it makes sense)? This book calls you "bootstrapped," even if you raise a bit of money.

Do you aim to avoid raising money from investors who want you to raise another round of funding every 18 months and consider it a failure if you don't sell for $1 billion? This book calls you "bootstrapped," even if you raise a bit of money.

Why Focus on SaaS?

You might wonder why this book focuses specifically on SaaS and not just tech startups, such as two-sided marketplaces, biotech, direct-to-consumer applications, and games.

Specific >>> General

Focusing on SaaS allows me to give extremely specific advice that doesn't apply to other types of companies. For example, I could spend pages discussing churn, onboarding, and lifetime value (LTV), topics that don't apply to some companies.

If this book were not focused on a single type of company, I'd have to take the typical approach and water down my advice, providing you with less actionable information and far less value.

A Repeatable Playbook

SaaS may seem boring to people outside of it. Instead of the next sexy direct-to-consumer viral game, ride-sharing app, or social network, SaaS focuses on solving pain points for businesses. Thus, the ideas seem pretty dry if you can't see past them and look at the exciting part: you are solving someone's problem in a real way and creating a valuable business while doing it.

Through my years of starting companies and hosting events for startup founders through MicroConf, I've been exposed to thousands of founders and their journeys, both successful and unsuccessful.

Several years ago, I began to pull out success patterns, turning them into SaaS-specific rules of thumb, best practices, and frameworks. Those have coalesced into a repeatable playbook that I've seen tested and improved in my own startups and in those I've invested in or advised.

This book is the culmination of those years of trial and error.

Why Is SaaS the Best Business Model?

There are a couple of reasons why SaaS companies have the best business model in the world. Let's dig deep into some of the most relevant.

Recurring Revenue

Every business wishes it could charge a subscription for its services and get paid in advance before it renders those services.

There's a reason every consumer product niche is now blanketed with subscription companies, whether they sell snacks, underwear, or Star Wars merch. The subscription razor startup Dollar Shave Club sold for a billion dollars.

John Warrilow wrote an entire book on how to do this with traditional, non-subscription businesses called *The Automatic Customer: Creating a Subscription Business in Any Industry.*

Subscription revenue is a business cheat code.

Recurring revenue protects you during recessions and builds on itself every month. Every business wants recurring revenue, but only some can pull it off.

With SaaS, it's built into the business model. People expect to pay you every month (or year) for your product. There is no need for financial gymnastics; you get it free with SaaS.

Recession-Resistant

During the 2008 financial crisis, I owned a small software product that sold for a $295 one-time fee. As the recession hit, my revenue dropped by 80% in one month. I was lucky I had other sources of income to support me, a few of which charged a subscription. This crisis was my wake-up call: recurring revenue handles economic downturns exceptionally well.

Even if revenue plateaus during a recession or other unexpected events (like the COVID-19 global economy shutdown), revenue tends to be insulated against a free fall thanks to its recurring nature.

Not Dependent on Luck

My mental framework for success involves varying degrees of hard work, luck, and skill. The amount of each depends on a myriad of factors, with hard work and skill being the easiest to control.

Launching a B2B SaaS company is about building a real product for real customers who pay you real money. You're solving a problem, and therefore your profitability is based on finding a problem that's worth paying to resolve and solving it in a way that makes your users desperately want your solution. There's not much luck in that formula.

Compare that to a crypto startup, a new social network, or a ride-sharing app. Each of these spaces has had dozens of companies focused on them for years, and their winner-takes-all dynamics require hard work, great execution, and a good bit of luck.

There were many social networks before Facebook and ride-sharing apps before Uber. Still, variables out of your control, like smartphone penetration, consumer behavior, or Internet speeds, can make or break your company. This tends to be different in SaaS.

Don't bootstrap a SaaS product if you want to start a startup for the excitement or the press coverage. Instead, move to a startup hub, raise boatloads of money, and hope you've bought the winning startup lottery ticket. If you want to start a company with a higher chance of a "base hit" outcome (meaning millions or tens of millions in revenue or enterprise value), go with SaaS.

Not Fighting a Battle on Two Fronts

One of the most challenging problems to solve when building a startup is kickstarting a two-sided marketplace, where you need to bring both supply and demand to the table simultaneously (think eBay or Uber). It's a bit like marketing two startups at once, and one of the most common pieces of advice I give to bootstrapped founders who want to start a two-sided marketplace is: Don't.

It's not that they can't work, but they usually need a ton of funding to get off the ground, and building one requires much more luck than most people realize. With a two-sided marketplace, if you have 10 people on one side and 1,000 on the other, your business is a failure. With B2B SaaS, if you have one, 10, or 100 customers, you have a business.

You Don't Need Funding

There are certain startups, such as those that manufacture physical products, marketplaces that charge a transaction fee, or social networks that push off monetization for years, that burn capital in the early days much faster than they generate it from customers. These businesses require outside funding to pay for their burn until they hit scale.

For the most part, SaaS does not have this capital requirement. There are thousands, if not tens of thousands, of SaaS companies that are profitable, continue to grow, and have never raised a dime of outside funding. Companies like Zoom, Slack, and PagerDuty went public with an enormous percentage of their venture capital still in their bank accounts. Actually, Zoom had more cash in the bank than they had raised!

As I said in the introduction, I am not opposed to outside funding, and in many cases, it's the best option to survive the early days and reach escape velocity. But one of the best aspects of SaaS is that you don't need funding, but if you find it helpful at some point, your capital efficiency and incredible margins make it relatively easy to raise.

High Profit Margins

Because of the low cost of servicing additional customers, SaaS companies reach gross profit margins of 90% and net profit margins of 50% or more at scale. This capital efficiency is one of the reasons they are easier to bootstrap than most other types of startups and an abundance of cash flows into SaaS, both from venture capital and private equity.

High Exit Multiples

The increasing volume of cash-seeking to invest in or acquire SaaS companies has driven their valuations higher and higher each year. Exact valuations depend on your annual recurring revenue (ARR), growth rate, platform risk, and other factors. Still, it's not uncommon for a growing SaaS company doing $1 million or $2 million annually to sell for four to eight times top-line revenue.

Note I didn't say four to eight times profit, a multiple many businesses would kill for.

Think about it this way: for every $1,000 of monthly recurring revenue (MRR) you generate, you generate $60,000 of value in your company (assuming you sell at five times annual revenue). That is an unreal multiplier on your effort.

You may build your company and run it forever, which is great. But if you decide to exit at some point, know that the value you've created in your company is enormous compared to the same size company in most other sectors.

Achieving Escape Velocity ↗

When I sat down to write this book, I sketched out a long list of topics. It quickly became apparent that topics relevant at one stage of a startup are less relevant at another. For example, if you've just launched, you should approach finding customers differently than if you're making $100,000 in MRR.

To that end, I decided to focus this book on topics that will help a business with some semblance of product-market fit take its company to the next level.

My mental shorthand for product-market fit is: *you've built something businesses want and are willing to pay for.* With SaaS, product-market fit starts very weak and often begins to strengthen when you're around $10,000 or $20,000 MRR. It then transitions to the next level, what I call escape velocity, once you've built one or more repeatable growth channels.

The process of moving from "a product people love" (product-market fit) to "a product people love, and you can find more of them every week" (escape velocity) involves building your product for your ideal customer, finding one or more marketing channels that work, building moats, reevaluating pricing, building your team, unlocking SaaS Cheat Codes, and finding and fixing bottlenecks in your funnel.

The strategies in this book are arranged into six parts:

- **Market:** How to understand your market and beat your competition
- **Pricing:** Structuring pricing for maximum growth
- **Marketing:** Finding and scaling the right marketing channels
- **Team:** How to get the right people working with you
- **80/20 Metrics:** The highest-impact SaaS metrics you should be watching
- **Mindset:** The psychological factors that can help you become a successful founder

Although all of the strategies, tools, and frameworks in this book will help your business grow, keep an eye out for four super-strategies, which I call the SaaS Cheat Codes.

These Cheat Codes—expansion revenue, virality, net negative churn, and dual funnels—can grow your business at an incredible pace if you get them right.

Let's dive in!

Market

Strengthening Product-Market Fit

As I said in the introduction, this book assumes you have some semblance of product-market fit, but it is a long road from launching a product to building something people want and are willing to pay for.

So much of your company's success will depend on developing a deep understanding of your market and ideal customer. Often, this understanding becomes a moat of sorts and is one of the best ways I know to strengthen the love your customers have for your product.

Although there are many ways to develop this understanding, one of the most reliable is having conversations.

Most entrepreneurs don't have enough conversations with potential, current, and past customers. It's time-consuming, and while running a company, you're doing everything from building new features to troubleshooting security issues to answering support emails. Even if you see the value of customer research, how do you find the time to do it?

There's another reason entrepreneurs don't talk to customers: fear—fear that they're bothering them, fear the customer will say something they don't want to hear, fear that it's a waste of time.

In my experience, the conversations you have with customers, whether via email, chat, or on a call, will be some of the most valuable time you spend understanding your market, especially in the early days.

These conversations will inform your product road map and how you build features as well as your positioning, marketing copy, and pricing. You'll build a better product for your ideal customer faster than if you try to guess what people need.

When Ruben Gamez, the founder of SignWell, decided to build an e-signature tool, he knew he needed to do something unique to make his product stand out. He talked to people about the problems they had with e-signature tools, and one thing he kept hearing was that customers wanted to send a link to the document that needed to be signed rather than having the email come from an impersonal third-party service.

Ruben listened, and that was one of the many features he heard through customer conversation that helped SignWell become a strong player in an incredibly competitive market.

Who should you be talking to?

- Prospects
- Customers
- People who decided not to become customers
- People who became customers and then canceled

Asking the Right Questions

The key to getting actionable information from customers depends on whether you're doing early customer discovery or researching how (and whether) to build a new feature. But typical questions might include:

- Can you walk me through a sample flow?
- What problem are you trying to solve?
- What do you currently use to solve this problem?
- What did you use in the past?
- What are some of your biggest frustrations about this solution?

As a general rule, ask open-ended rather than leading questions. If you say, "We're thinking about building this. Check out this mock-up," most people will have a hard time being honest. They don't want to hurt your feelings, so they go along with it and you don't get useful data.

Deep dives into customer conversations have literally filled books. For a great resource on doing customer interviews as a founder, check out the book *Deploy Empathy: A Practical Guide to Interviewing Customers* by Michele Hansen. You can also listen to my interview with Michele on Episode 586 of the *Startups for the Rest of Us* podcast.

In these conversations, this is the time to put your consultant hat on and have a conversation entirely focused on your customer's needs, not your product—this part is important.

As Jim Kalbach, author of *The Jobs To Be Done Playbook,* told me on Episode 577 of my podcast, when you look at the people you serve through the lens of your own solution or product, it clouds your judgment.

❝ *By taking yourself and your offering out of the equation, potentially —there's no guarantee of this—you can actually find opportunities* *that you wouldn't see otherwise.*"

Building the Features Your Customers Want

One of the popular memes in the entrepreneurial world is that your customers will tell you what they want—just launch something ASAP, and they'll tell you how to make your product better.

But that is assuming your customers know what they want.

Henry Ford never actually said his customers wanted faster horses instead of a car, but the wisdom in that story is strong enough that people have been repeating it for 100 years.

Your customers don't know how to build software as well as you do, and even if they did, they'd never match the insights about the market you've learned building your product. In fact, if they could, you should probably close up shop, go to bed, and try something new in the morning.

As my last startup, Drip, began to grow, we received several feature requests per day, and we had to constantly make difficult decisions about which ones we were going to invest our limited time and resources into building.

I wanted to build a product that people used—a lot of people—and that meant learning how to take certain points of feedback to heart and discard the rest. A big part of the reason Drip was successful was that we used the feedback from our best customers to find and solve problems other email service providers didn't.

This involves learning how to separate valuable ideas from distractions, and until you hire a product manager (something that typically happens north of $1 million in ARR), you're the person who needs to decide which features will strengthen your product-market fit.

I filtered them by putting them into three buckets: The Crackpots, No-Brainers, and In-Betweens.

The Crackpots. First up are the crackpot requests: ideas so far out of left field you can't imagine why they want it or even understand what they mean.

The crackpot suggestions are the easiest to process because you know right away you're not going to build them.

Here are some examples of oddball requests:

- Features that would require building an entirely new product (e.g., "Why can't I use your email provider to publish blog posts?")
- Features that would turn your product into a clone of your competitors (e.g., "It'd be great if you could add a shopping cart, and a CRM, and lead scoring, and . . . just build a clone of Hubspot and charge me one-tenth of the price.")
- Features that are the opposite of your product's strengths (e.g., "I like that your UI's so streamlined, but can you add these ten new options to fit my rare and unique use case?")

No, no, and no—these folks make it easy to say, "Sorry, these requests are not a fit for us."

No-Brainers. You'll also get requests that are either already on your road map or make you wonder why you didn't think of them. Once in a while, your customers will come up with an idea that's so awesome you want to put it into production immediately.

These are also easy to handle because you know they'll objectively improve the product and make it more valuable for your users.

For instance, at one point, a customer reached out to see if there was a way to retroactively add a tag to his subscribers based on links they had clicked in the past.

Well, no . . . but it was a great idea and not terribly hard to build. Drip was (and might still be) the only product that offered that feature, and it became a powerful tool for our users that they couldn't use anywhere else.

List pruning is another example of a no-brainer feature. Subscribers zone out over time, driving up the cost of your email plan and skewing your metrics with "dead" users that wouldn't open an email if it contained the cure for cancer.

Most list management systems don't have built-in list pruning, and it's a big chore to go back and figure out which subscribers to delete from your list.

So when a customer reached out and asked if we could make a button that would prune inactive subscribers according to some basic best practices, we immediately put it on our road map.

In-Betweens. I'd estimate only 10% to 15% of the feature requests are crackpots, while maybe 20% are no-brainers, so that leaves a lot of judgment calls.

The reality of running a software product is that you will get dozens of feature requests that aren't necessarily bad, but aren't slam dunks either.

You can't build all of them—that's how good software bloats into a mass of buttons, boxes, toggles, and settings. As the founder, you're the gatekeeper. You're going to have to say "no" to an awful lot of good ideas.

I usually roll my eyes when people quote Steve Jobs as their reason for taking a particular course of action because he was such an outlier that his thinking won't work for most of us. But on the topic of saying "no," he nailed it:

> "*People think focus means saying yes to the thing you've got to focus on. But that's not what it means at all. It means saying no to the hundred other good ideas . . . Innovation is saying 'no' to 1,000 things.*"

As we just discussed, one way to weed through feature requests is to figure out the problem your customer is trying to solve. When a customer asks you to add a new kind of button, they don't really care about the button—they're trying to do something in the software they can't figure out and believe a button is the best way to accomplish that.

Your job is to figure out the problem your customer is trying to solve, not just build the solution they suggest. Then, figure out whether that solution requires a new feature and whether it's worth building.

It's pretty easy to do this in a way that makes people feel like you've listened to them, whether you end up building what they want or not.

Ask yourself these three questions:

Question #1: What's the Use Case for This? Or, in layman's terms, what problem are you trying to solve?

You can get at this by asking questions such as: "What leads you to want that? What problem are you trying to solve with this feature? What are you currently using to get that done?"

Sometimes you'll realize you already have a way to solve the customer's problem but in a way they hadn't discovered yet. At that point, it's up to you to explain how they can use your tool to accomplish their goal and maybe add something to the UI to help other users find it more easily.

If your product doesn't already meet the customer's needs, you must determine whether you can build the feature and whether you want to. At the end of the day, this depends on how many people you think will use it. This leads us to . . .

Question #2: What Percentage of Customers Will Actually Use This Feature? More than 5%? Ten? Twenty? You won't know for sure, but spitballing your best-guess percentage can help you decide whether to build the feature and how prominent it should be in the UI.

If you think only 5% to 10% of your customers will use something, go a step further and spot-check which users they are.

If it's a random group of users, it might not be worth building. But if that 5% to 10% are power users and this feature would make your product more useful to them, consider building it and keeping it hidden from the average user by omitting it from the standard UI and only enabling it upon request.

Why hide these features? Because if the vast majority of your users will never need them, adding dozens of checkboxes and dropdowns will make your core product confusing for them.

If you think 20% or more of your customers will use a feature, it's something to consider building.

But first, you have to ask yourself one more question . . .

Question #3: Does This Fit with My Vision of the Product? Every feature has opportunity costs. Every hour you spend building a feature is an hour you don't spend building a different one.

Ultimately, you're the founder. If a feature request doesn't fit with your vision of what the product should be, it's probably not one you should build.

One elegant solution to many of those "big lift" feature requests is to add integrations. Problems that might take weeks or months to solve are often already solved by another product. If you can integrate their application programming interfaces (APIs) into your product in just a few days, fulfilling those requests can be a win-win for you and the customer.

Too many products get cluttered with obscure features that only help one or two customers, and too many entrepreneurs get bogged down in building endless feature requests instead of focusing on making great software.

As a bootstrapper, time and money are precious, and feature requests can wipe out both quickly.

But the most important resource in your company—and the most scarce—is your vision for your product and how it solves your customers' problems.

Solving your customers' problems is the road to strengthening product-market fit. Expect a long, slow road of customer conversations and hard decisions with incomplete information as you work to build something people want and are willing to pay for.

How Can I Compete in a Competitive Market?

A mature, competitive market can be a total bloodbath, but it can also be a fantastic place to grow your business once you get a foothold.

For one, a competitive market is proven. You already know that people are willing to pay for the problem your product solves. If you can figure out a way to stand out among the crowd, you could be looking at a very lucrative business.

For another, it can be a great place to find an underserved niche of customers who are frustrated with what the big incumbents have to offer.

You've heard stories of small startups disrupting the status quo, but usually they're led by an experienced founder, or they're backed with impressive amounts of funding. I don't recommend starting your first startup in a highly competitive market without prior experience or funding—or both.

But if you're getting a foothold in a mature market and you're ready to scale up, here are some ways to do it.

Compete on Price

Competing on price is tricky, but you can get traction if you offer more than 80% of the product for half the cost.

In a mature market, the large players have pricing power that allows them to raise prices relatively frequently. When you don't have the brand or credibility, it can be hard for customers to justify paying the same price for your service.

In addition, your competitors' pricing power often leaves plenty of room for you to underprice them and still have significant profit margins.

In this case, if you've built a product that's easier to use than the large players' products and you are less expensive, it can be an incredible one-two punch when convincing early adopters to switch.

I'm not recommending that you remain the low-cost provider indefinitely, but before you've built a strong brand, this can be a good strategy. Of course, you should keep adding value to your product and raising prices as it matures.

Compete on Sales Model

Another place to innovate against bigger competitors is with your sales model.

Many incumbents in competitive markets have a high-touch sales process where customers must schedule a demo before learning the price. They may also have mandatory setup fees.

If you enter the space with a low- or no-touch sales process and your pricing is listed on your website, you can start winning customers who don't want to jump through so many hoops.

When we launched Drip into the marketing automation space, one of the biggest players, Infusionsoft, wouldn't let you see their product until you had paid a couple-thousand-dollar setup fee and gone through the required training.

With Drip we offered a free trial—all you needed was a credit card. This made early adopters more willing to give us a shot.

Compete on Product

The third area in which you can outmaneuver larger competition is your product—especially if you're dealing with larger companies who have 10- or 15-year-old code bases and clunky user interfaces.

In the world of accounting software, many Quickbooks refugees are looking for a solution with better UX or more streamlined features. Xero was able to get a foothold by building a cloud-first accounting package.

In customer relationship management (CRM), plenty of people are looking for a Salesforce alternative that feels fresher and like a better fit for their business. Pipedrive, Close, and many others have been able to carve out pretty great businesses by improving product usability and ease of setup.

If you can enter a competitive market with modern UX and the ability to ship features quickly, you can build a great product while also developing the reputation of being innovative and nimble.

This is a temporary advantage. Over the years, your UI and code base will age, and you won't be able to ship as quickly. By that time, you should have a more mature product that handles the majority of use cases versus the early days when you're using any advantage you have to compete against full-featured products.

How to Market Against Large Competitors

In Judo, you're taught to use your opponent's weight and strength against them. The same holds true in business.

You're not going to outmuscle or outspend entrenched, well-funded competitors. You need to use their biggest strength (their size) to your advantage.

With big companies comes slowness to react to the market, dated UX, legacy code, and all the competitive disadvantages we discussed. If they've been dominating the market for years, they probably also have a group of users who have defaulted to using them but are frustrated by the experience.

If you can find this group that desperately wants to escape and help them migrate to your tool, not only will you gain new customers, but they will rave about you to others in their circle.

You've turned the strong brand of your competitor into a massive lever: word of mouth.

People also love the scrappy new underdog, which can be a great marketing angle for a startup in a competitive market.

Rental car company Avis did this with its "We're #2, we try harder" ads in the 1980s.

I did this with Drip. Our home page essentially communicated that we were “not Infusionsoft, Marketo, or Pardot.”

The headline on our homepage was “Lightweight Marketing Automation That Doesn’t Suck,” which was very much a swipe at our larger, clunkier competitors.

How Much Should I Worry about Competition?

Most founders worry too much about their competition. They follow them on social media, scour the news for updates, and have Google alerts set up on competing founders and executives.

And it’s not doing their mental health any favors.

You do need to have your finger on the pulse of the competitive landscape, but for your own sake, you need to be deliberate about how you keep tabs on your competitors. It can lead to a negative emotional cycle if you start every day focusing on the competition while having your bagel and coffee.

In my experience, unless you’re losing deals to specific competitors on a regular basis, it’s more helpful to keep your eyes on your own paper. Your number one goal should be serving your customers—not perseverating over your competitors’ every move.

There are only two things you should care about when it comes to competition:

High-Level Updates. If an announcement is big enough to merit space in your industry news sites, it’s probably worth paying attention to. Is your competition announcing new funding? Launching big features? Making a major shift in positioning?

Read it and understand it. But rather than losing sleep over whether you should follow suit, spend that energy studying what that tells you about your industry and your customers.

Deals You're Losing. If you're losing deals to a competitor, you should figure out why. Is it because you lack specific features? Is it because of your pricing? Do they have SOC 2 compliance?

Once you understand why you're losing deals, you can decide how to address those objections. Maybe you need to carve out a new position in the market where those features aren't as necessary. Maybe you need to revamp your pricing or look into securing similar certifications.

Conversely, here's my rule of thumb for what you should ignore about your competitors:

Low-Level Details. From the outside, it might look like your competitors have their act together. But if you're not in the room where decisions are being made, all you see is a polished image. Don't let that polished image fool you into thinking your competitors know what they're doing.

I've seen many, many companies raise millions in funding and then misprice or misposition its product, move too slow on the technical side, make a mess of its branding, or make any number of first-time founder mistakes.

Just because your competitor is acting, don't assume it's the right move—or even a well-thought-out move.

Their Funding. If a competitor raises funding, that's not necessarily a bad sign for you. It's not a sign they know what they're doing—just that they were able to convince an investor that there's an opportunity.

When I see a company raise funding, the most common result is that it blows through it in about 18 months. If it doesn't have significant traction by then, it is likely to flame out.

Funding obviously works in some instances, and if you have an experienced competitor gaining traction and adding funding to the mix, that's something to take to heart.

As a caveat, if your competition raises a lot of money—like $5 million or more—be aware they might be trying to suck the air out of the market by either offering a free solution or dropping prices.

Being Copied. If your product is successful, competitors will copy you.

Even if you don't say it to your team, even if you never say it publicly, it's infuriating to see someone come into your space and copy your positioning, features, or marketing. It's even more annoying when they pretend like they're innovating by doing so.

As a founder, it's your job to manage your mindset and not let their plagiarism derail you. It's inevitable in business, especially if you're on the leading edge of your industry.

Sometimes copying gets so bad you need to enlist the help of a lawyer to send a cease and desist. Things have to be pretty blatant to get to this point, but it happens.

One way to make yourself less vulnerable to copycats is to build a moat around your business.

How Can I Build a Moat?

As you scale your company, you need to think about how to proactively defend against competition. The more success you have, the more your competitors will grab their battering ram and start storming the castle.

In medieval times, you'd dig a moat to keep enemy armies from getting anywhere near your castle. In business, you think about your *economic* moat.

The idea of an economic moat was popularized by the business magnate and investor Warren Buffett. It refers to a company's distinct advantage over its competitors, which allows it to protect its market share and profitability.

This is hugely important in a competitive space because it's easy to become commoditized if you don't have some type of differentiation.

In SaaS, I've seen four types of moats.

Integrations (Network Effect)

Network effect is when the value of a product or service increases because of the number of users in the network. A network of one telephone isn't useful. Add a second telephone, and you can call each other. But add a hundred telephones, and the network is suddenly quite valuable.

Network effects are fantastic moats. Think about eBay or Craigslist, which have huge amounts of sellers and buyers already on their platforms. It's difficult to compete with them because everyone's already there.

In SaaS—particularly in bootstrapped SaaS companies—the network effect moat comes not from users, but integrations.

Zapier is the prototypical example of this. It's a juggernaut, and not only because it's integrated with over 3,000 apps. It has widened its moat with nonpublic API integrations, meaning that if you want to compete with it, you have to go to that other company and get their internal development team to build an API for you.

That's a huge hill to climb if you want to launch a Zapier competitor.

Every integration a customer activates in your product, especially if it puts more of their data into your database, is another reason for them not to switch to a competitor.

A Strong Brand

When we talk about your brand, we're not talking about your color scheme or logo. Your brand is your reputation—it's what people say about your company when you're not around.

Having a strong brand means you're in a lot of conversations. When people discuss options, you're in the mix. People are talking about your product on their podcasts and forums and to their colleagues. You've developed a reputation for reliability and innovation, or maybe you've become known for solving a unique problem.

Once people recognize and trust your name, you become a brand rather than a commodity. Instead of comparing your pricing and features to everyone else, prospects will start looking at you as a unique offering, even if your features are mostly equivalent to a competitor.

There are hundreds of CRMs, but I bet if you and I sat down and tried to name every one we could think of, we'd top out at maybe a half dozen. Those are the ones with strong brands. It's a significant advantage.

A quick note: Positioning is part of your brand.

Maybe your digital asset management software could work for anyone, but if you decide to focus your product on museums and make sure it talks to the collections management tools they're already using—and that your marketing and website are speaking the language of that world—you may be able to build a brand moat around a segment of the market that can be tough for competitors to cross.

Positioning is a book in and of itself, so I'll point you to *Obviously Awesome* by April Dunford as one of the best positioning books for SaaS companies.

Owned Traffic Channels

A friend of mine owns a SaaS company that's competing in a massively crowded space. His product gets 500,000 unique visitors a month because he's exceptional at search engine optimization (SEO), and his company ranks on the first page of Google for many high-volume terms.

He owns these organic traffic channels in his market, so even though other names on those pages might be more recognizable, he can stay highly competitive.

Even if you own a high-traffic search term on Google, Amazon, or the WordPress plugin store, you can have a pretty commoditized product that can still succeed.

One caveat is that this moat can be a bit dicey to maintain because the algorithms at any of those companies can change quickly—and have. Google's many updates have tanked businesses overnight that depended solely on SEO-driven traffic.

High Switching Costs

Products that require a significant amount of work to migrate away are said to have high switching costs. High switching costs reduce your churn and create a moat that keeps customers from switching to a competitor simply because that competitor is newer, cheaper, or even builds a better product.

Most APIs are difficult to leave because to do so requires expensive developer time to integrate with a new product. Companies like

Stripe, Twilio, and SendGrid have a pretty hefty switching cost moat.

Tools like Slack are difficult to switch from because of the need to obtain buy-in from every manager in an organization. Also, because of the high number of integrations pushing data, Slack requires effort to recreate.

Tools with low switching costs are those in which history is mostly irrelevant, and the time it takes to recreate something you've built in the tool is low or nonexistent.

For example, a social media scheduling tool is easy to switch from because there is no critical history stored or complex workflows that need to be recreated using a new tool.

Likewise, one-click SaaS analytics tools that tie into your Stripe account are relatively easy to switch from because they are "one-click easy" to set up.

False Moat: Unique Features

As makers, we want to believe that features are what differentiate us. After all, we've spent hours building and perfecting those features. That's worth something, right?

Don't get me wrong. Unique features are a fantastic differentiator—for a few months, until a competitor duplicates them.

You can create a brand around continuously building and shipping innovative features, especially when your competitors are legacy companies who don't have your feature velocity. Believing these features in and of themselves are a moat, though, is a trap.

By definition, a moat should endure; it should sustain itself and grow stronger as your business grows. If you're relying on a constant stream of new features to differentiate you, you don't have a moat. You have a hamster wheel of features.

You can absolutely find growth that way. But realize you'll also want to focus on some of the other, more enduring, moats.

Should I Translate My Product into Other Languages? (And Other Common Mistakes)

I've seen founders fall prey to some common siren songs, which I talk about below. With some rare exceptions, heading down the following roads is a distraction at best and a major blunder at worst.

Founders often take one of these paths to solve a real problem: their company isn't growing fast enough. In most cases, the solution is not to pursue these distractions but to strengthen your product-market fit or improve your marketing and sales efforts.

If your company isn't growing, you don't need to chase a siren song. You need to get to the root of the real problem, one that's not usually fixed by the approaches discussed below.

Translating Your Product into Other Languages

When a founder comes to me with the plan of translating their product into another language, I advise against it.

The mechanics of translating a web application aren't complicated. But you're not just translating the app—or even your knowledge base, documentation, and website. You need to translate your marketing into Spanish or German or French. You need someone on your team who can do email support or live chat in that language. You need to manage your social media presence, maintain an email list, and many other things to support customers in that language.

There has only been one time when a founder told me they were going to translate their app and it turned out to be a good decision. In that case, he wanted to add Spanish as a second language (English being the first).

In this case, the users of his product primarily spoke Spanish, but their managers—his actual customers—mostly spoke English. This founder didn't have to take on the challenge of handling marketing or customer support in Spanish, but by translating the app itself, he made it more accessible to his customers' end users.

White Labeling

White labeling is when another company pays you to license your product and present it with its own branding, and the moment you launch a successful product, people will start emailing you with "exciting opportunities" to white label.

For the most part, these conversations are a big waste of time. Usually what you have is someone who wants to start a business but can't build their own product. They want to pay you per account they add, but they have no audience or distribution. In the end, you'll spend a bunch of time talking to them, writing up contracts, and taking feature requests—for nothing.

If you're approached about white labeling by a large player, it's worth having the conversation. You know they're not wasting your time because they don't want to waste their own.

To justify the effort of white labeling, I recommend charging an upfront fee. We're talking tens of thousands of dollars—$30,000 to $50,000 at a minimum. If someone balks at paying that fee, they're not willing to put enough skin in the game to warrant your efforts.

I'm not a fan of white labeling in most situations. It's a way to serve customers and make money without building a brand. We discussed above how a brand is a moat, and losing that is an unfortunate consequence of white labeling.

Adding Other Verticals Too Early

Adding other verticals is the easiest siren song for me to justify. Sometimes it makes sense to pivot or expand to other niches. For example, if you have a product that's working well for wedding photographers, chances are it will also serve wedding videographers.

But unless you're pivoting your product, you must be extremely careful about adding new markets. For example, if you add wedding coordinators to your wedding photography SaaS, you're probably dealing with audiences that have different needs—even though they're in the same industry.

The danger of adding other verticals flippantly is that it can lead to a lot of complexity in your product. Unless you already dominate a particular niche and are moving into another or are making a full-blown pivot into the new space, be careful with this one.

Underpricing Your Product

I dig into pricing in the next chapter, but I want to call out what is perhaps the most common mistake I see founders make: listening to the voice in their head telling them that keeping their price low is the key to kickstarting growth. It's not.

Most of the reasons I see founders pricing too low are psychological, not logistical. You're afraid of rejection. You're having trouble seeing value in your product because you built it. You're comparing your price to cheap competitors rather than seeing how much value you bring to your customers.

Pricing too low holds your business back in two major ways. First, if you're charging $10 instead of $100 a month, you have to find 10 times as many customers. Second, you'll have a much harder time finding those customers because you'll be severely limiting the marketing channels you can afford to use.

Pricing incorrectly can mean the difference between building a $250,000-a-year business and a multimillion-dollar business. We'll dig into that in the next chapter.

Pricing $

How Should I Structure My Pricing?

Pricing is the biggest lever in SaaS, and almost no one gets it right out of the gate.

Fortunately, you don't need a PhD to structure your pricing well. Like most things in SaaS, finding the right pricing structure is one part theory, one part experimentation, and one part founder intuition.

I wish I could tell you a single "correct" structure, but it varies based on your customer base, the value provided, and the competitive landscape.

Most founders price their product too low or create confusing tiers that don't align with the value a customer receives from the product.

On the low end, if you have a product aimed at consumers, you can get away with charging $10 to $15 a month. The problem is at that price point, you're going to be dealing with high churn, and you won't have much budget to acquire customers.

That can be brutal, but if you have a no-touch sign-up process with a product that sells itself, you can get away with it. Castos's podcasting software and Snappa's quick graphic design software are good examples of products that do well with a low average revenue per account (ARPA).

You'll have more breathing room (and less churn) if you aim for an ARPA of $50 a month or more. In niche markets—or where a demo is required or sales cycles are longer—aim higher (e.g., $250 a month and up).

If you have a high-touch sales process that involves multiple calls, you need to charge enough to justify the cost of selling it. For example, $1,000 a month and up is a reasonable place to start.

If you're making true enterprise sales that require multiple demos and a procurement process, aim for $30,000 a year and up (into six figures).

One of the best signals to guide your pricing is other SaaS tools, and I don't just mean competition. Any SaaS tool a company in your space might replace you with, a complementary tool or a tool similar to yours in a different vertical can offer guidance, but make sure you don't just compare features; compare how it's sold.

As mentioned above, the sales process has tremendous influence over how a product should be priced.

There are so many SaaS tools out now that a survey of competitive and adjacent tools can give you a mental map of the range of prices you can charge.

No matter where your business sits, one thing is true: "If no one's complaining about your price, you're probably priced too low."

Segmenting Your Customers

The first step to structuring pricing tiers is to segment your customers. You need to figure out who's using your product, how they're using it, and what value they're getting from it.

For example, I use a company called SquadCast to record interviews for my podcast (disclosure: I'm also an investor). SquadCast users come in all sizes, and each segment has a range of needs and price sensitivities.

You can imagine a hobbyist podcast aimed at fly-fishers or gamers would pay $10 or $15 a month but not much more. This person would have a consumer mindset.

In the middle of the market, a podcast like mine that's aimed at the entrepreneur audience will probably be okay paying $50 to $100 a month because it'll get at least that much value out of its show.

At the top end of the spectrum, you can imagine large podcast networks like NPR or I Heart Radio have much larger budgets to spend on recording—and they also have very different needs than a hobbyist or mid-market show. They'll be happy to pay 20 times more than the base plan because of the sheer value they'll get from the product.

SquadCast's tiers reflect this by scaling up the usage and features they offer in each subsequent tier.

SquadCast

Indie	Pro (Most popular)	Studio	Enterprise
Emerging creators	Larger production teams	Agencies and small businesses	Large businesses
$20 USD PER MONTH	**$40** USD PER MONTH	**$80** USD PER MONTH	**Contact Us** PERSONALIZED QUOTE
- 5 Hours Audio Recording Per Month	- 12 Hours Video Recording Per Month	- 25 Hours Video Recording Per Month	- Volume Video Recording
- 1 Show	- 3 Shows	- 10 Shows	- Volume Shows
- 2 Team Member Seats	~	- 10 Team Member Seats	- Volume Team Member Seats
~	- 5 Team Member Seats	~	~
- 1 Integration + Zapier	- 2 Integration + Zapier	- Unlimited Integration + Zapier	- Unlimited Integration + Zapier
- Iso & Mix Audio Tracks	- Iso & Mix Audio Tracks	- Iso & Mix Audio Tracks	- Iso & Mix Audio Tracks
	- Video + Screen Recording	- Video + Screen Recording	- Video + Screen Recording
		- Master Audio w/Dolby Voice	- Master Audio w/Dolby Voice
			- ACH Invoices

When you segment your customers by size and usage, you start to see how your pricing tiers work to offer the most value to customers while driving growth for your business. Getting pricing tiers right also allows you to tap into the SaaS Cheat Code we've already alluded to: expansion revenue.

SaaS Cheat Code: Expansion Revenue

Expansion revenue is when customers pay you more as they get more value from your product. They might manually upgrade to the next tier or be auto-upgraded as their usage changes.

There are two basic ways to build expansion revenue into your pricing tiers.

Value Metric. A value metric is how your company measures the per-unit value of your product.

A typical value metric might be the number of users your customer brings to your product. MailChimp's prices are based on subscriber count, so subscribers are their value metric. Salesforce's prices are based on the number of seats, so seats are their value metric. Usage is another good value metric, like the number of recording hours for SquadCast or gigabytes used for Dropbox.

This works because the more your customers use your product (and the more value they get out of it), the higher the price they will pay.

If you design this so that the value metric is tied to your customer's success, they will be much less resistant when moving up to the next level. They've increased the number of subscribers on their mailing list, hired more people for their sales team, or increased the number of podcasts they put out every month. Jumping up to the next tier means their own business is growing alongside the value they're getting out of your product.

A quick note about seats as a value metric: Seat-based pricing is a common value metric for SaaS, but I recommend offering it only if two users from the same company see different things when they log into your product.

For example, if User A logs into the CRM and sees tickets assigned to them by User B, seat-based pricing makes sense. If both users log in and have the exact same experience, they might as well just share a login.

Feature Gating. The second way to unlock expansion revenue is by expanding the features your customers can access at higher plans. This tends to be less effective than value metrics because it's not as intrinsically tied to the growth of your customers' businesses.

Feature gating can be useful if you find your customers don't necessarily experience an increase in usage as their company scales but do end up needing an expanded set of features.

Gated features can be ones that you build or integrations you have with other services. For example, if a customer wants to export their data to Tableau, you automatically know they have the budget to pay for an expensive SaaS product. It's reasonable to assume this type of customer would see enough value to pay a premium for a Tableau integration.

In this case, choosing to pay for higher tiers still provides them with increased value linked to their own success.

Using Both. You can also combine a value metric and feature gating. I tend to encourage people who are in the early stages to start by using just one model, then refine that as they get to know their customer base and what they need. But in many cases, having per-seat pricing with two or three levels of feature access is not a bad way to go.

Be aware, though, that it can quickly get complicated when you try to use both feature gating and value metrics.

Enterprise Pricing

One big mistake founders make is not charging enough for their most valuable customers: those on enterprise plans.

Typically founders will undercharge, especially if they come from a development background and aren't used to paying enterprise-level prices for software. If you worked in a big company but were never part of a sales conversation about which $50,000-a-year software to purchase, those numbers might seem unreal next to your $7-a-month GitHub subscription.

With enterprise deals, you're not charging more just to charge more. There are key differences in the way enterprise customers buy products. There are extra hoops to jump through in their procurement process. You might need to do a custom integration. If your enterprise tier is underpriced, you'll lose money trying to sell to enterprises and service their accounts.

A loose rule of thumb is to charge 10 to 20 times more than your standard plan. If you are only charging two or three times more, you won't be able to hire that customer success person or salesperson you need to manage the high-touch sales process, let alone cover the cost of acquiring enterprise clients.

Should I Offer Freemium?

According to the State of Independent SaaS Report, 66% of SaaS products offer a free trial, but only 17% offer a forever-free (aka freemium) plan.

Is freemium the path for you? It depends. Freemium is like a samurai sword. If you know what you're doing, it's powerful. If you don't, you can cut your arm off.

There's a reason that many venture-funded companies offer freemium and many bootstrappers don't. This is because venture-backed businesses can afford to support a large freemium user base.

For a bootstrapper, though? You typically need as much cash as you can get in the short term. Freemium pushes your revenue into the future, so if cash is tight it can be less than ideal.

Free plans don't just magically work—you need to understand how they function.

Freemium works best with a simple product that doesn't require much work for the customer to get value, where the support burden is reasonably low and there's some level of virality built into the product.

For example, an e-signature app is pretty straightforward. The user uploads a doc, then sends it or signs it within minutes to get near-instant value. If you've created an easy-to-use product with a good UI, you will not have to do much in the way of support or onboarding. Plus, every time a user needs a document signed, they're spreading the word about your product because you've built in the Virality Cheat Code (covered later in the book).

As long as your per-user or per-usage costs are pretty low (i.e., you're not paying much every time someone sends an email through your e-signature app), then you can offer a freemium tier without spending much cash.

On the other hand, a freemium model would likely not work for a construction management software CRM with low to zero virality and a complex setup and onboarding process.

You can run into problems if your free users are in a completely different market than your paid users. Do they need additional features or require more support? Are they in different verticals or categories?

If your freemium users aren't at some level helping push growth of your paid tiers, offering a freemium plan isn't the right call.

Competitors and Freemium

If all your competitors offer freemium, does that mean you have to? Not necessarily.

The reality is that many startup founders don't actually know what they're doing. They'll launch a freemium option for a while and then shut it down because it was too much work or didn't provide the results they expected.

If the majority of your competitors offer a free plan and it's become a sticking point with the prospects you talk with, then you might consider offering a freemium option to remain competitive.

Or you could come into the market like automated customer acquisition software Bounce Exchange (now Wunderkind). Not only did they not offer freemium like the majority of their competitors, but they also launched at $2,995 a month. Some people in the community thought they had lost their minds.

However, Bounce Exchange had a different business model. They became successful because not only did they offer software, but they also managed the entire service for their customers. They did a great job of creating a product for a very specific segment of the market who was willing to pay for it rather than racing for the bottom with pricing.

Should I Ask for a Credit Card Up Front?

One of the most common questions I get about trials is whether or not you should require a credit card up front.

Dropping the credit card requirement is an attractive option because you can get ten times as many trials if you don't ask for a credit card. Although I typically default to asking for a credit card up front, it helps to understand the pros and cons.

Entering a credit card is a qualifying event. A person willing to enter a credit card has more interest in your product than someone who won't, and it reduces the number of tire kickers in your sales funnel.

This is important because all those free trials can actually be pretty expensive. If you're bootstrapping, you probably don't have the funds or time to manage support and onboarding for people who are not likely to convert.

When thinking through requiring a credit card up front, ask yourself:

- How high is my support burden for each customer going through the trial?
- Do I have the bandwidth to multiply my support burden by 10, knowing there will be many more tire kickers?

A lot of first-time founders tell me they don't want to require a credit card because they want to get as many people as they can into the app to learn what people want.

The problem with this approach is you end up getting a bunch of noise from users who aren't your core customers. Instead, you end up with an unfocused group chattering about different things. You don't know who to listen to, making it hard to focus on the next feature you should build.

When You Should—and Shouldn't—Ask for a Credit Card

Of course, there are always exceptions.

If your product is often adopted by employees who don't have company credit cards, you may want to drop the requirement.

A good example of this would be Slack, Trello, or Dropbox, which have freemium levels and trials that anyone can set up. Team members can set up an internal Slack group without having to get approval from a manager with a company credit card.

Once your initial champion and their team start using your product at work, they'll see how valuable it is (and probably want to add integrations). Then, they can make the case to their manager to break out the company credit card and upgrade to the pro version.

If you're not in that situation but are having trouble getting people to sign up with a credit card, the answer probably isn't to drop the requirement. Instead, make sure you've built something people want and learn how best to market it.

Opening the Floodgates

As I said, my default recommendation is to require a credit card up front. But I've seen founders decide at a later date to open the floodgates by removing the credit card requirement. Here's what these founders had in common:

- Their companies were early but established.
- They had more than $20,000 MRR.
- They knew their customers and market intimately.
- They knew their conversion, churn, LTV, and other metrics by heart.
- They had the resources to handle the influx of new trials.

Castos was one of these companies.

Founder Craig Hewitt didn't see a significant increase in his company's growth rate after Castos dropped the credit card requirement for its free 14-day trial, but overall the move was positive for the company. It enabled more employees from large nonprofits and organizations to try the product without having to ask for the company credit card number and gave Castos's sales team the ability to have prospects try the product more easily.

If your business has matured to a place where you know that if you had five or 10 times more trials you'd see increased growth, try dropping the credit card requirement for one to two months to see how it impacts the business.

But during that time, be obsessive about the numbers. Every metric you know by heart—trial-to-paid, churn, referral, etc.—is going to change once you pull the credit card requirement. Best case, you'll know within a month or two if it's working, but it can take several months because you also want to take into account changes in downstream churn. Craig said it took almost two months after Castos dropped the requirement for him to get a bead on the new numbers.

As I mentioned, your support requirement will also change when you add five to 10 times the number of new trials. You'll have to keep a sharper eye out for spammers and be much more active to convert users who don't provide a credit card up front because 90% of them wouldn't have signed up otherwise.

Just remember the golden rule of experimenting: Only change one variable at a time.

If you're going to remove the credit card requirement, don't also increase the price. Don't also introduce a free plan. Let the metrics stabilize so you can identify the patterns before you do the next thing.

You may want to consider a shorter trial length so you can see the results of your experiments even faster. A seven-day trial means you can run through four cohorts of trial users in the same amount of time as a 30-day trial—that's four times as many tests.

Not every industry or app can have a seven-day trial, but the shorter you make your trial—including having no trial at all but just a refund policy during the first 30 days—the faster you can experiment.

You also want some positive time pressure to encourage users to start getting value from your product right away. If you offer a 30-day trial, how many users will wait until you send out that final "your trial is ending" email to actually start using their account?

When Should I Raise Prices?

I recommend revisiting your pricing every six to 12 months because if you're like most founders, you're probably charging too little.

Most founders think that the biggest reason to raise prices is to make more money, and that's a good reason to do it. More money means you can hire more people, spend more on marketing, or take home more profit.

But there is a second-order effect that most people miss. Across all SaaS marketing approaches (which I talk about in the next chapter), maybe five are available if your ARPA is, say, $20 a month. If you charge $500 a month, it's closer to 10. And if you charge $5,000 a month, you have every SaaS marketing approach at your fingertips.

So it's not solely about making more money. It's about providing you with more options to grow your business.

It's not technically hard to raise your prices. You change a number on the pricing page and make an API call, and there you go. You doubled your prices. It's the emotional part that typically makes it hard for most founders.

Some tell me, "If I raise my prices, I'll make my customers angry. I'll completely crush my business. I'm going to take it too far." Or they'll push back by saying, "I can't just raise prices to infinity, right? At some point I'll get too high for my market or where my product is at."

Yes, you will probably make some customers angry. No, you're probably not going to crush your business—and you can always roll back pricing if you find out it's a mistake. It's possible to raise your prices higher than what the market will bear. But more founders err on the low side than the high side regarding pricing.

When you talk to customers who are churning or prospects who aren't converting, you'll inevitably hear from someone that your product is too expensive.

Dropping your price is rarely the answer when it comes to product-market fit because pricing is such a huge lever in the business. But those customers might still be right.

It may be that you are overcharging for what your product currently does. When I launched Drip, it was just an email capture widget that sent out an email sequence. I charged $49 a month and kept getting feedback that it was too expensive.

I could have dropped the price to $19 or $29 a month, but I didn't want to build a cheap tool. I wanted to build a tool that was an easy sell at $49 a month.

I call that "aspirational pricing," and I used it to push me to keep improving the tool until it was worth what I wanted to charge for it.

A lot of founders underprice their products because they don't want the rejection. They don't want to hear people complain.

Some people are going to complain about your pricing no matter what—but it's incredibly tough to build a business when you're underpriced. Instead of dropping your price, one approach is to figure out what you need to build to make your tool worth what you are charging.

How to Raise Prices

That said, there are ways to raise your prices that cause the least amount of shock to your customers.

One simple option is to decrease the value metric of your plans but keep the prices the same. If today, customers get 3,000 subscribers for $49, switch it so tomorrow they get 2,500. Your pricing in dollars stays the same, but people need to upgrade to higher tiers faster.

Another option is to hide your lowest pricing tier on your pricing page. This should be relatively quick to hide, and you'll see the impact of sign-ups relatively quickly.

The typical way to raise prices is to increase them across the board and watch your numbers to see how new customers react.

Or you could multiply your pricing by 10 to go after a new segment in your market. One of the first batches of TinySeed companies was a SaaS tool for interior design teams called Gather (Gather-It.co). It wasn't growing as fast as it could in its early days for a couple of reasons. First, it was catering to one- and two-person architect and interior design shops, which are naturally price-sensitive. Second, its price point was only $29 a month.

It decided to pivot and go upmarket. This was a gamble because the features a 40- or 50-person architecture or interior design firm needs are much different than small shops, and Gather had to build to get there.

It took nearly 18 months of painful work, but eventually it was able to double its prices, then double them again. It churned out lower-end customers as it landed bigger ones, and now it charges ten times what it did at the beginning.

Most of the time, your price increases won't be as drastic as Gather's. We'll talk about the logistics of raising prices in a minute, but first I want to discuss two potential mindsets when raising them.

Experiment vs. Certainty

There have been times when I raised prices because I knew we had to do it, and I didn't care if it hurt our funnel in the short term. We were going to raise prices, and things would have had to go pretty far sideways for me to roll it back.

But there have been other times when I was less certain, and I proceeded with a bit more caution.

Knowing which time is which is a question for your "founder gut." If you're not certain, treat it like an experiment. Make the change one you can undo quickly if you need to, then watch it painstakingly every day.

Give it a couple of weeks, or maybe a month, and keep asking yourself: What are the results? How many people are coming through my funnel? What does my trial-to-paid look like?

This is what we call the "poor person's split test." In a perfect world, you'd split test the pricing evenly at the same time—but that's difficult for a SaaS company. I only know of one that's done it.

Zapier didn't publish pricing on its public page. When you clicked the sign-up button, it was forked within the app so that half saw one price and half saw the other. That gave Zapier great data, but for the rest of us, monitoring cohorts of sign-ups at the new price will have to do.

If you know for certain that you're going to raise prices, make this a marketable event—especially if you'll be grandfathering in current users at the old rate. Announce the price increase ahead of time and let the world know that if they've been meaning to try your app, now's a good time. If they sign up for a trial now, they'll be grandfathered in at the old price.

Grandfathering Existing Customers

If you're going to raise prices on your pricing page, what do you do with your existing customers? Do they keep the old pricing? Do you upgrade everyone?

In a perfect world, you'd just upgrade everyone. After all, if you have a restaurant and change your prices, everyone who shows up next week to get a burger and fries pays the same new price. With SaaS, though, it's a bit trickier.

Why consider grandfathering old customers?

- If you're worried people might go to a competitor
- If you're worried thousands of customers will write in angrily to your support people
- If you're worried you'll damage your brand and your reputation

How should you decide?

Use Rob's Rule of 10: If raising prices for existing customers will not grow MRR by at least 10% (ideally more), it's rarely worth considering.

The amount of headache, support burden, brand damage, and potential churn is so onerous that if you're only going to grow by a few percentage points, it's best just to grandfather those users in.

On the other hand, if you're changing your pricing significantly—say, from $9 a month to $99 a month, it's not economically viable to grandfather the existing customers. Prepare yourself for the onslaught of emails as your lower-paying customers churn out to make way for higher-paying ones.

Here are a couple of tips:

- Never promise to grandfather customers for life. Next time you raise prices, you might not want to. Or you might sell the company and the new owners don't want to honor that deal. Lifetime deals are for one-time sale stuff, not for SaaS.
- Enterprise customers expect annual increases of 5% to 10% as standard, so build that into your contracts.

Raising Prices Well

If you're not going to grandfather in existing customers, give yourself a comfortable amount of time between announcing price increases and actually implementing them.

I recommend two to four months, depending on how difficult it is to switch away from your product. Too short, and people will feel trapped. Too long, and when the price hike does come around they'll have forgotten and think you're raising prices again.

One thing you never want to do is raise prices on existing customers without sending notice. That's a recipe for angry customers.

When you announce a price increase, use this template:

1. Set the stage for the value your product offers (i.e., we've been around for some time, we've become a trusted provider in this space, etc.).
2. "We're changing our pricing." Let them know up front what's going on.
3. Provide high-level justification about why you're changing your pricing (i.e., we've added tons more value, we're in a completely different space than when we launched, we're expanding our features, etc.).
4. (Optional) Offer more specifics about whom it impacts, when price increases will go into effect, etc.
5. (Optional) Provide more justification if you feel it's necessary.
6. "Reach out with questions." Let them know your doors are open for questions, comments, and feedback.

For a solid example, check out CartHook's announcement of its price increase from a few years ago (bit.ly/carthookpricing). Its announcement covers each of these six basics really well.

Another great example of how to raise your prices is Gymdesk, a TinySeed company that provides online management software for fitness and wellness businesses.

Founder Eran Galperin knew his product was underpriced at the beginning of 2021. In launching the company six years prior, Eran had priced Gymdesk below his competitors. No longer wanting a reputation as a budget product, Eran decided to raise Gymdesk's pricing considerably, in some cases by more than 50%.

In an interview for the MicroConf YouTube channel, Eran told me about the emotional aspect of raising his prices. He'd built a personal relationship with many of his small business customers and knew how thin their margins were. He wrote an email explaining the price increase, then opted to grandfather in old customers at their original rate for a few months before moving them to a discounted plan.

"Most of our customers responded very positively to that," Eran recalled. "Some of them even wrote back and said, 'Congrats, you deserve a raise.'"

In the end, only a few of Gymdesk's more than 600 customers left as a result of the price increase. The company's MRR increased by 25%, MRR growth went up by around 70%, and its ARPA has continued to climb.

With the increased revenue, Eran was able to be more aggressive with marketing and hire four full-time team members to grow his one-person operation.

Marketing

How Do I Find More Customers?

There's a harmful myth among aspiring founders that you can build a great company simply by building a great product. Do that, and it will "sell itself."

This myth is perpetuated by successful companies—think of Apple or Basecamp—where it appears their products *do* sell themselves. But really, they are so good at marketing that it's invisible to most people. Think about it. Many of us tune into Apple's product announcement live streams, which are essentially hour-long commercials.

Yes, you can get lucky with early adopters and word of mouth. Some people do. But the vast majority of the time, if you're going to be successful, you have to build a good product *and* market it well.

And if you're going to grow your SaaS business to seven figures, you have to figure out how to spread the word without pouring millions of venture capital dollars into glossy launch parties and billboard ad campaigns.

As a founder with a technical background, I understand the tendency to shy away from marketing and sales, hoping the business will catch on based solely on the virtues of the product.

I did that for years until I realized the only way to reliably build a business was to get good at marketing.

I know for a fact marketing is a huge area of concern for most SaaS startups because when we ask in the TinySeed application what a company's biggest hurdle is, the vast majority say they need more customers.

Many people can write code. But creating a product people want and actually selling it is relatively hard. To build a seven- or eight-figure SaaS business, you need to develop your marketing tool belt.

Can't I Just Hire Someone for That?

At some point, you'll reach the stage where you have enough revenue to hire a marketer. But you'll still need to learn marketing well enough to get to that place.

Marketing strategy is hard (and expensive) to hire out, and you'll need to know enough about marketing to understand whether or not the marketer you hired is doing anything useful. Marketing strategy is all about deciding which approaches to try in which order, tracking the return on investment (ROI) of those approaches, and building on them to grow the business. It's a difficult role to outsource.

Once you have a marketing strategy, it's possible to hire someone to implement each piece of it. You can hire a freelancer or agency that specializes in pay-per-click (PPC) or SEO. You can hire copywriters, social media managers, and graphic designers.

But at a minimum, founders need to learn SaaS marketing strategy or risk losing control of and visibility into a crucial part of their business.

Let's take a quick look at the basics, then talk about scaling from there.

Marketing Funnels

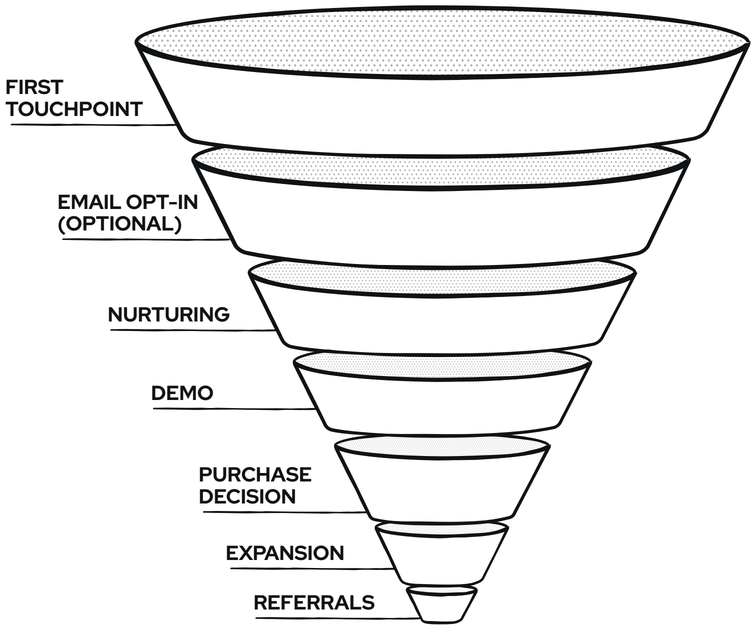
A marketing funnel describes the stages of your customers' journey from the moment they learn about your business to the point at which they become paying customers.

The most common marketing funnels in SaaS are high-touch, low-touch, and dual. Which one is right for your product? That depends on your market, pricing, and customer base.

As Aaron Kassover, founder of AgentMethods, told me: "*It seems like most SaaS founders want to go low-touch because that's what we prefer as customers. We hate having to talk to a salesperson before getting our hands on a product. But most markets prefer high-touch. Switching from low- to high-touch was really hard for me because it wasn't what I wanted, but it's what propelled us to product-market fit. How you sell is as important as what you sell.*"

High-Touch Funnel

A high-touch funnel gives your customers a lot of human interaction as they make their buying decision.



For example, a prospect might first become aware of your product because they stopped by your booth at a trade show and met with you or a member of your sales team. They volunteer their contact info and become a lead.

Leads aren't necessarily going to buy anything. They might have just been interested in the whitepaper or webinar you were offering. A lead becomes an opportunity when they express interest in your product.

For example, when someone from your sales team follows up with an offer to jump on a sales demo, they say yes. If the demo goes smoothly, that opportunity might sign up to become a new customer, who usually gets passed to a customer success representative to set up their account.

Obviously, the higher touch your marketing funnel is, the more money you need to make from each customer. High-touch funnels work best for companies that charge a minimum of $500 a month and focus on customers who can afford that price point.

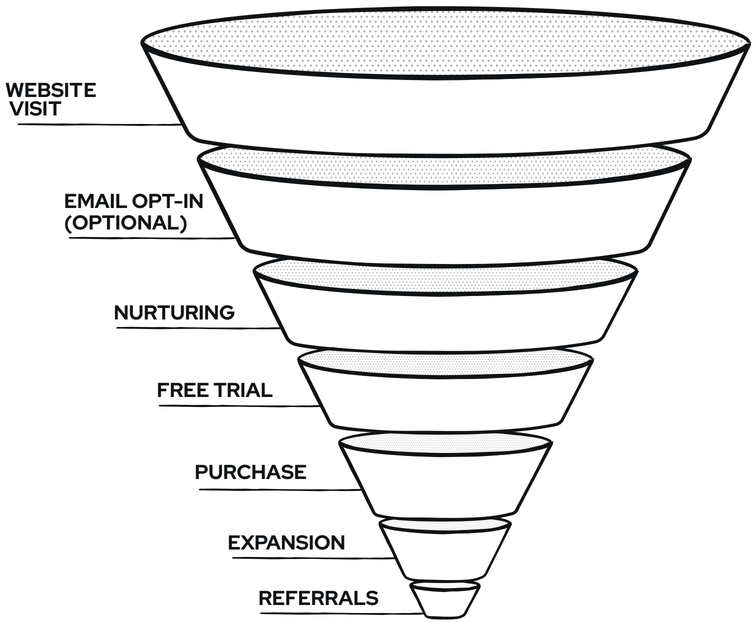
Higher-touch funnels also tend to focus more on outbound marketing tactics like cold-calling, cold-emailing, and LinkedIn outreach.

The percentage of conversions you'll have with a high-touch funnel will vary depending on your industry and pricing. A product with a $100,000 annual contract value (ACV) will have a very different close rate than a product with a $3,000 ACV.

You can do a lot to improve your funnel metrics by bringing in qualified leads, educating them before they become opportunities, and closing the sale during demos. In the pre-acquisition days of Drip, we had a close rate of 25% to 35% from demo to customer, and most were one-call closes.

Low-Touch Funnel

A low-touch or no-touch funnel works when you have a wide market and a low-priced product.



In this model, you set up a large funnel where leads come in on their own and are led automatically through a sales process without much input from you or your team.

For example, your prospects become aware of your image-editing app through ads on Facebook and sign up for your free seven-day trial without talking to someone on your team. After the trial is up, some percentage automatically converts to customers on your $19-a-month paid tier.

Imagine your basic sales funnel:

- You drive traffic through content marketing, SEO, appearing on a podcast, etc.
- They hit your website to check out your product.
- They sign up for a trial or request a demo.
- They use your software during a trial period, then convert to paying customers.
- They stick around as happy, satisfied customers.
- They eventually upgrade their account or refer additional people to your product.

Because you're not spending as much energy qualifying leads, you'll see lower conversion rates and usually higher churn—but because your low-touch funnel doesn't require as much hand-holding, your cost to acquire a customer and the cost to sell to that customer are very low.

A low-touch funnel tends to be a volume game. It is highly focused on inbound marketing tactics like SEO, PPC ads, content marketing, and influencer marketing.

Rules of thumb for funnel metrics change over time, so instead of including them here I have a page on this book's companion website where you can see my most up-to-date numbers: saasplaybook.com.

SaaS Cheat Code: Dual Funnels

Another Cheat Code is something I call a dual funnel, which targets both a wide audience at a low price point and a premium or enterprise audience at a high price point.

This works well because the wide, low-touch funnel allows you to bring in customers who can help your product spread via word of mouth. That brand recognition can feed your high-touch, high-priced funnel.

One example is SignWell, an e-signature app with a wide, low-touch funnel bringing in professionals and small businesses who only need a few documents signed. On the flip side, that same e-signature app might target mortgage brokers who need 5,000 or 10,000 documents signed each month.

What often happens in the case of a dual funnel is that in the early days the low-touch funnel brings in the majority of the revenue. As you build your user base and bring in more high-end users, they will become a higher percentage of your revenue base.

At that point, when you've developed a strong brand, you can build the business on both fronts, and your growth will be dramatically accelerated.

What If My Funnel Isn't Working?

You'll know your funnel isn't working if your numbers at any phase are below the typical rule-of-thumb ranges. Customers and prospects are spilling out of your funnel somewhere, and you need to fix that leak before you start pouring more money into marketing.

Start by examining the metrics at the bottom of that funnel, then work your way up to see where the problems are.

- How happy are your customers?
- How long do they stick around?
- How many of your trials convert to paid customers?
- How many website visitors sign up for a trial?
- How well does each traffic source earn you website visitors?

If your customers are frustrated and churning out after a few months, it doesn't matter how many people you're sending to your website. Everything at the top of the funnel is a vanity metric when the bottom of your funnel isn't healthy.

Keep in mind that you need to have enough traffic coming through your funnel that reliable patterns appear. If you're only tracking a handful of customers, you won't have enough data to see noticeable patterns.

How you fix your funnel depends on where the problem exists. Starting at the bottom of the funnel:

- Are people churning out in high numbers after paying for a month or two? You likely have a problem with onboarding or product-market fit.
- Are people trying your product, but not many are converting to paid? This could be a problem with your onboarding (you're not getting them to experience value quickly enough) or your sales process.
- Do you have a lot of website traffic, but no one is signing up for your trial? You probably have an issue with your value proposition, marketing copy, or positioning.

Having an outside perspective can be invaluable in helping you troubleshoot the problems in your funnel. Ask members of your mastermind for help, reach out to an advisor or mentor, or even consider hiring a consultant to help plug the leaks before you spend too much time and money bringing in new leads.

Business-to-Business SaaS Marketing Approaches

Diving into the implementation details of marketing approaches is beyond the scope of this book (I'll leave that to books like *Traction* by Gabriel Weinberg and *Hacking Growth* by Sean Ellis and Morgan Brown). But I want to include a list of the 20 most common business-to-business (B2B) SaaS marketing approaches so you have an idea of where to start.

Then in the next section, we'll dive into my framework for filtering and prioritizing marketing approaches to find the best options for your particular situation.

The "Big 5" SaaS Marketing Approaches

I call these the big 5 because they are the most common marketing approaches that work for B2B SaaS companies.

SEO. SEO is not just about Google. Even though it's the largest search engine in the world, there are many more search engines people use and which might be best suited to your product.

For example, YouTube is number two, and Amazon is in the top five. There's the iOS App Store, the WordPress Plugin Repository, the Shopify App Store, the Adobe App Store, and hundreds of other places worth learning how to get your product to rank. Some of these in particular might be where your ideal customer is searching for your solution and have the benefit of being fairly easy to rank.

With Drip, we built a couple of WordPress plugins and ranked well for email marketing-related terms in the WordPress Plugin Repository.

I've seen founders build Chrome plugins to rank in the Chrome Web Store. Or write a Kindle book using the expertise they developed running their company to rank in Amazon's search.

PPC Advertising. PPC advertising is the tried-and-true, usually expensive way to drive traffic quickly. This is where you buy traffic through keywords on Google (AdWords) or in platform-specific marketplaces like Facebook, Instagram, YouTube, Amazon, Capterra, LinkedIn, and many others.

Cold Outreach. This is outreach to prospective customers directly via email, phone, LinkedIn, or DMs on social media. This works best when you can get a list of businesses that need what you've built and can find a signal that they're at a point in time where they need your product.

For example, at Drip we knew Infusionsoft customers would be looking at new options as their one-year contract came up, so we got a list and pinged them about eight months after their contract started to see if they were interested in exploring other options.

Integration Marketing. This is one of the only marketing approaches that also serves your current customers because it improves your product by adding more features through integration with a partner.

To come up with ideas, think about what steps happen before or after your product. Contact a potential partner and outline what you can offer them by integrating with their platform. Don't build the integration unless you both agree to promote it. Promotion can include:

- Publishing a blog post
- Emailing both partners' email lists
- Posting a tweet
- Writing a knowledge base article
- Mentioning your company in-app
- Hosting a webinar
- Listing products in an add-on marketplace

Approach this in an agile way. Start by building a minimum viable integration. Maybe the first version of the integration doesn't have gold-plated authentication, and instead users have to paste in their keys. However, the first version should be enough to launch the integration and promote it to see if it brings in new leads.

Then, determine which integrations your users are using the most and circle back to release another version that maybe adds OAuth or a simpler integration.

The results of these integration marketing efforts can be long-lasting because many times the articles, forum posts, and other assets produced by the company you are partnering with will be high-converting and produce leads for years.

A related marketing approach is pursuing partnerships. These are similar to integrations, but you can execute them without writing code. Many partnerships are simply a reciprocated promotion of the other product to your audience using one or more of the promotional pieces we discussed.

Content Marketing. Although it is often coupled with SEO, content marketing relies solely on virality or on building an audience slowly over time, without the long-term benefit of organic search.

This includes writing blog posts you hope will make it to the top of social news sites like Hacker News and Reddit and building a media brand alongside your product (something I recommend only for very well-funded companies). It also includes producing content to educate people at different steps in their funnel whom you already have permission to contact.

Most people think of blog posts when you mention content marketing. However, content can include books, ebooks, audio (think podcasts), video (think YouTube), or even in-person courses that are given away to bring links, traffic, and leads and build credibility.

Most founders start by producing the content themselves, then hire people to help with production later.

Other (Still Important) Marketing Approaches

Amazing companies have been built on these approaches, but your ability to use each one depends on the availability in your niche and the cost to execute compared to your ACVs.

Affiliate Marketing. Affiliate marketing is a common way to tap into established audiences in your industry. You'll typically pay between 10% and 30% commission to the affiliate. If you're selling to large companies, affiliates are usually replaced with resellers.

This is almost more of a business development approach than a marketing one because you typically have to build or have a relationship with the influencers who have an audience. Setting up an affiliate account and posting a link isn't enough to get your affiliate marketing program up and running.

The danger of affiliate marketing is how much affiliate payouts can eat into your revenue.

At scale, most SaaS companies have net margins between 30% and 50%, where a 50% margin is impressive. If you're giving away 30% of your top line to affiliates, you can have an incredible revenue graph but have the least profitable SaaS application on the block.

One way to be cautious with your commission rates is, for example, to give 20% to most affiliates and provide higher rates (say, 30%) to premium affiliates who have large audiences.

In-Person Events and Trade Shows. Events and trade shows can be a fantastic marketing approach, depending on your industry.

Maybe your product serves an industry where the majority of business is still done in person at conferences and trade shows. To earn buyer trust, you need to meet them where they are. One company I know of is marketing to city governments, and the founder has had success meeting prospective clients at in-person events.

Attending an event may also be a brand-building exercise where you immerse yourself in the community and establish strong relationships with your customers.

Whatever the reason, take a hard look at your ROI. Will landing one enterprise deal give you a return on the cost (in money and time) of attending the event? Then it could be worth it. But if you have to sign up 20 new customers to make your money back, you either need to raise your prices or in-person events are not a fit for your business.

Free Tools (i.e., Engineering as Marketing). Usually tied in with SEO, this approach involves building a software tool you give away for free to generate traffic to whom you pitch your main product. Examples include HubSpot's famous website grader and SignWell's e-signature creator.

Hangouts. These are gathering places like online forums, private Slack groups, Facebook groups, and subreddits.

Q&A Sites. This approach focuses on answering questions on Quora, Stack Exchange, and other relevant Q&A sites.

Virality. Make sharing your product with other potential users part of your product's experience. (We discuss virality in the metrics chapter when we look at my 3 High/3 Low Framework.)

Other People's Audiences. This involves putting yourself in front of established audiences through guest posts, podcast tours, and YouTube tours.

Daily Deal Sites. These sites promote SaaS products and other software tools at deep discounts. For example, AppSumo and PitchGround.

Launch Sites. These sites curate and showcase new SaaS and software products for clients to discover. They rate them by a voting system. For example, Product Hunt and BetaList.

Lesser-Used Approaches

These are tactics used by such a small percentage of mostly bootstrapped B2B SaaS companies that we won't focus on them in subsequent sections (but I want to list them here because they might be valuable in certain situations).

Building an Audience/Community/Movement. Five percent of companies funded by my SaaS accelerator, TinySeed, built an audience before they launched. And only a handful more, at most, build one as they scale.

Display Ads. This is buying ads based on a cost per impression rather than cost per click.

Viral/Stunt Marketing. Stunt marketing works for Red Bull, but it's expensive, and the brand-building that results doesn't have a measurable ROI.

Speaking Engagements. Although it is not a very scalable marketing approach, putting yourself on stage as an expert in front of 200 or 2,000 potential customers can provide solid ROI and will inevitably strengthen your brand.

PR. PR can help build buzz about your product in the media (either mainstream press or news sites specific to your industry).

Offline Ads. These include billboards, bus stop benches, and radio ads.

How Do I Know Which Marketing Approaches Fit My Business?

As we saw in the previous section, there are many ways to market your product. And a big challenge we face as founders is prioritizing which approaches to try. This section discusses how to filter and prioritize marketing approaches to find the handful that are likely to work best for your business. Here's the framework I recommend for this:

The Three Factor Framework

The three factors of a marketing approach are speed, cost, and scalability. You need to take all three into account when choosing marketing approaches.

Speed. How long does it take for a marketing approach to start returning results? Weeks? Months? Years? How long can you afford to spend money and time on a marketing approach before you start to see new customers?

In the early days, faster approaches are the key to getting initial customers. As your product matures, you can afford to spend more time cultivating slower marketing approaches.

Ideally, you should be working on a fast and a slow approach at the same time. If you focus only on fast tactics, you'll always be chasing new leads. If you focus only on slow approaches, you'll end up in a wasteland while you wait for your efforts to show results.

For example, you can get relatively fast results with cold outreach while building organic search results that can sustain you in the long term.

Cost. How much will it cost? In the early days, you'll want to think in terms of hard costs—dollars and cents—instead of your own time.

The higher your ACV, the more money you'll have to invest in marketing. As you scale your product and raise prices, you'll notice that more marketing approaches become available.

Scalability. Can you scale this approach to reach more people?

Many marketing approaches in the low-scalability camp are either one-time events like posting to Product Hunt or require a lot of time, like answering questions on Quora.

On the other hand, if there is enough search intent for your tool and you can dial in search and PPC ads, you can often keep turning that dial up to reach more people without spending a lot more time.

Which Approach Works Best at Your Price Point?

It's challenging to create a concrete list of which approaches work best at what price point because many of the costs vary widely depending on the space you're in, whether the founder is running the effort, or whether you're trying to hire outside staff to assist. But I've put together this list to serve as a starting point.

Note that any approach listed as a fit for a particular ACV can also be used in the higher ACV tiers.

High ACV (~$25,000 and up)

- Cold-calling (fast, scalable)
- In-person events (moderately fast, less scalable)
- PR (moderately fast, less scalable)
- Offline ads (moderately fast, scalable)
- Display ads (fast, scalable)

Medium ACV (~$7,500 and up)

- Cold-email (fast, scalable)
- PPC (depends on cost; fast, scalable)
- Integration marketing (slower, somewhat scalable)
- Free tools (you might get lucky and have it work with low ACV; slow, scalable)
- Hangouts (fast, less scalable)
- Q&A sites (fast, less scalable)

Low ACV

- Content marketing (moderately fast, scalable)
- SEO (slow, scalable)
- Virality (slow, scalable)
- Affiliate marketing (only if founder-managed, otherwise move to Medium ACV; moderately fast, scalable)
- Partnerships (only if founder-managed, otherwise move to Medium ACV; can sometimes be fast, somewhat scalable)
- Other peoples' audiences (fast, less scalable)
- Launch sites (fast, less scalable)
- Daily deal sites (fast, less scalable)

Prioritizing Marketing Approaches

Once you've narrowed down your list of potential approaches, you can use any prioritization framework to sort them. I've found the ICE framework helpful for this purpose. ICE stands for:

- Impact: If this works, how big will the potential impact be?
- Confidence: How likely is this to succeed?
- Ease of implementation: How easy is this to execute?

ICE is often used to prioritize feature development, but it's also a good tool for prioritizing marketing.

To do this, list potential approaches in a spreadsheet and rate them on a scale of one through 10 for each of the above characteristics. I've seen people use several methods to get the score.

1. Score = Impact x Confidence x Ease: This gives you a score with an exponential impact. In other words, the higher you rate any one area, the more confident you need to be.
2. Score = (Impact + Confidence + Ease)/3: This gives you an average of these three scores.

However you rank those facets, using the ICE framework is a way to get your approaches into a spreadsheet and figure out which are the best to start with. You can list things by high-level approaches (content marketing, PPC) or by individual tactics (ebook, blog post, guest posting, YouTube ads, Facebook ads).

You can also start by ranking high-level approaches, then start a new tab in the spreadsheet to break down the top approaches by individual tactics. Then tackle the highest-rated approaches and tactics first.

Tips for Running Marketing Experiments
Once you have prioritized the marketing approaches that best fit your company, it's time to put them into practice. Every change you try in your marketing strategy will be experimental until you verify that it delivers the results you are aiming for. Before you start running your first experiment, keep these three tips in mind.

Keep a Marketing Changelog. I recommend keeping a marketing changelog, a chronological record of everything you try, even small things like updating marketing copy on your website. At some point in the future, if you realize trial conversions dropped 14 days ago, you can refer back to your changelog to see how you broke it.

Measure. One mistake I see founders make at this stage is that they don't measure the effectiveness of each channel. They buy ads but don't track how much it costs to acquire each new customer based on that ad. They do SEO but don't attribute where the traffic is coming from.

Track as you experiment, and don't rely on your gut instinct of whether a marketing approach is working.

You'll want to track your cost (in dollars and time) and the results. Compare the results of each marketing approach with your initial rating in the ICE framework. How close were you in your original estimation? What assumptions did you start with, and how correct were they? What issues did you uncover? What did you learn?

This data will help you hone your founder gut when choosing your next marketing approaches and tactics with the ICE framework.

Don't Try Too Much at Once. The approaches that are best suited for your business will vary, but there is one thing that tends to work well: doubling down on a successful approach rather than spreading your energy all over the place.

You don't need to try dozens of marketing approaches. I've seen SaaS companies grow to seven and eight figures by picking just one or two marketing channels and scaling them well, whether that's content marketing, SEO, integration marketing, PPC, display, outbound, or something else.

You want to start by testing one approach that works quickly (e.g., cold outreach) and one that works slowly (e.g., Google SEO). Assuming you have the resources to tackle both, this allows for short-term and long-term growth. Obviously, the more resources you have, the faster you can move.

Getting a Head Start On Marketing Experiments

Solving the puzzle of how to reach your audience can be tough if you're a trailblazer. But if you're fighting for the attention of a defined audience in an established market, here's a way to get a head start: Talk to the people who are already marketing to them.

If you've built a tool for developers, you can take a look at other tools your audience would likely use. If they're not direct competitors, contact the founder and ask for a quick call. Say, "I know you market to developers. Could you spend 30 minutes chatting with me about what's working for you?"

Obviously, if they're a competitor they're not going to tell you their marketing secrets. But I've found that if they have a complementary or unrelated tool, they're often happy to help. That conversation might even turn into a partnership later.

You can also learn from your competitors. Watch how they're talking to your shared audience (with the caveat that just because they're marketing a certain way doesn't mean they know what they're doing).

You can also talk to their past employees to get the inside scoop. I've reached out to former employees of competitors on LinkedIn to say, "I've built a competitor to your former employer, and I'd love to chat about what's working in our industry when it comes to marketing." I make it clear they don't have to tell me any secrets they don't feel comfortable sharing, and I've even offered to pay them for their time.

Salespeople especially love to network—and talk. They're often a fount of knowledge about what sales brochures look like, how the marketing team works, and what your audience is looking for.

A final approach is to interview your competitors' founders and marketers. It's surprising how much you can learn by listening to a founder talk through their approach on a podcast.

Scaling Marketing the Smart Way

One of the biggest mistakes I see venture-funded companies make is raising buckets of money, then dumping it into marketing before they have product-market fit. They're driving a bunch of people to a product that few want or are willing to pay for.

It's not sustainable, and good marketing only makes a suboptimal product fail faster.

If you're a bootstrapper, you can't afford to make this mistake. Before you start pouring money into finding more leads, you'll want to do three things:

1. Make sure you have reasonably strong product-market fit.
2. Make sure your conversion, churn, and other bottom-of-the-funnel numbers are good.
3. Experiment to find the most effective marketing approach for your product.

What about Word Of Mouth Marketing?

You might have noticed that I left a big marketing channel off the above list: word of mouth.

Whenever I ask an entrepreneur where most of their leads are coming from and they say word of mouth, I like to keep digging. Usually, the real answer is that they don't know where customers are coming from.

Which is pretty dang risky.

Word of mouth comes when you start to have an identifiable brand—which generally happens once you get to about seven figures of revenue. You'll know you're there when the conversation moves from people talking about "marketing automation" to mentioning two or three players in the space, and you are one of them.

With my SaaS company, Drip, we had excellent word of mouth, but I could see what percentage of our new users came from integrations, organic search traffic, podcasts, and other channels. With Drip, as far as we could tell, word of mouth drove 15% to 25% of new users.

The takeaway is that word of mouth does not account for as much as you think it does—which means you need to get to work figuring out where your traffic is *actually* coming from.

If you don't know where your customers are coming from, how can you find more of them? How can you figure out which of your marketing experiments are working? How can you dial in your marketing budget better? How can you know which levers to move to scale your business?

It's easy to figure out where people from your PPC ads and SEO are coming from. It's much harder to figure out where direct traffic is coming from. You might have gone on a podcast and mentioned the URL. Someone might have read about it in a book, newspaper, or magazine. They might be on a different device than the one on which they originally heard about your product, so the referral link is cleared. This could be a returning visit.

You're not going to be able to attribute 100% of traffic, but you should try; any attribution is better than none. One simple approach is to ask customers how they first heard about you or ask where they heard about you right before they signed up. Both data points are helpful.

Another clever approach is the approach Ruben Gamez, founder of SignWell, takes. He told me, "When someone first visits your website, set a cookie with their referral source and another for the landing page. Then save those to your database when they sign up. You can also send them to Stripe, ChartMogul, Mixpanel, etc. for easy segmenting in those tools."

How Should I Structure Sales Demos?

(I know that sales isn't marketing, but I've included it in this section for those of you with high-touch funnels. If you have a low-touch funnel, you can skip this section and move on to the next chapter.)

I'm not a great salesperson. But even if you don't think of yourself as a salesperson, every founder needs to be able to sell—whether you're talking to potential customers, trying to raise investment, or even just "selling" the company's vision to new hires.

Many founders I talk to shy away from sales because they have negative perceptions. They think of the high-pressure salesperson at the car lot or the bored upsell from the customer service rep at their internet provider.

With B2B SaaS, sales shouldn't be sleazy. Instead, it should be an educational conversation.

My TinySeed cofounder Einar Vollset says, "*When selling SaaS, think of yourself as an unpaid expert who's helping the prospect solve their problem using software.*"

You're not trying to force a fit between your software and your prospect's problem. You're putting on your consultant hat to help your prospect define their problem and come up with a good solution.

Thinking of yourself as an expert problem solver first sets a good tone for sales demos. When I used to do sales demos, I would introduce myself as the founder and say, "I'm not trying to talk you into anything. I'd just love to show you our tool and get your feedback on how it might fit your needs."

If your tool doesn't fit their needs, it's far better to let that prospect move on (maybe with a recommendation for a tool that's a better fit) than to pressure them into signing up. Don't waste time or money onboarding someone who's just going to churn out after a month or two.

Qualify before You Demo

There are few things worse than showing up to a sales call to find out the person doesn't have the budget or the need for your product. As someone with intimate knowledge of customers who buy your software, you should have a good idea of the common threads that link them.

Asking even a few questions about budget, timeline, and the problem they are trying to solve can be a window into whether it's worth your time to jump on a demo.

Have a Script

Even though as the founder you can run a demo with your eyes closed, if you have a standard script, you are always ready to train someone new to take over sales.

Say No to People Who Aren't a Fit

If you know someone will not get value from your product or believe they will be a problem to support, do not be afraid to let them know you don't think they are a fit and recommend competing tools. If you are qualifying people in advance of your demo, this shouldn't be something you have to do often, but forcing a sale only to have a customer churn out a few months later will waste a lot of resources.

Follow Up

It's amazing how many salespeople don't bother following up. People are busy, and following up until someone tells you they are no longer interested is the process that good salespeople follow.

If Possible, Take Payment over the Phone

Giving a demo and hoping the person signs up through the website is not a great way to close a deal. Not every SaaS application is a one-call close, but if yours is, set up their account at the end of your call, including taking their payment info over the phone.

Ask the Right Questions

Here are good questions to ask during sales demos:

What Problem Are You Looking to Solve? Your demo is not a product tour; it's proof you can solve their problem. Focus on the customer's pain and how you can fix that.

When I was selling Drip and a prospect said, "I want to be able to email my website visitors, prospects, and customers different content," it was an easy demo.

Sometimes your prospect may be unable to clearly articulate the problem they're trying to solve. Or they may come in thinking they're trying to solve one problem when they actually need help with something else. The more conversations you have with prospects, the better you'll be at consulting with them.

How Are You Solving That Problem Today? Finding out what other tools and software they're currently using to solve their problem helps you frame your solution relative to what they're already familiar with.

For example, if they're using Mailchimp, I could compare Drip's features to what they already know. If they say Infusionsoft, I know they're used to a more expensive and clunkier tool. If they say they have an Access database and they're sending emails through a hacked-together system, then I could start by talking about the benefit of using an email service provider at all.

How Big Is Your Organization? Are they a solopreneur with 500 people on their email list? A company with a few dozen employees? A Fortune 1000 company with an intense enterprise sales process?

Knowing this can help you understand how valuable this sale is to your company and how much value you'll be able to provide to them.

How Sophisticated Is Their Understanding of This Category? Try to ask tactfully how much expertise they have so you can speak to their level. Again, when you're thinking like an expert problem solver, your goal will be to educate your prospect. They should walk out of the sales demo better able to do their job—regardless of whether your product was a good fit.

One way to ask about their expertise is by going deeper into how they're using other tools. For example, if a prospect tells me they're using Mailchimp, I could gauge their sophistication by asking if they've used any of the automation features.

What's the Decision-Making Process? Toward the end of a sales demo, it's good to get a sense of who else needs to weigh in on this decision. Are you talking to the decision maker? Does your prospect need to go back to a group? Is there any collateral you can provide, like a PDF or a guide, that will help them champion your product to their organization?

Talk through the timeline of when you can expect to hear from them, when you should follow up, etc. Then, when the demo's over, make sure they're in some type of CRM so that you follow up at the right time.

Demonstrate How Your Tool Solves the Problem

Thinking of yourself as a problem solver first helps solve another common issue I see with sales demos. Instead of giving prospects a painfully detailed explanation of every feature, only show them the relevant parts.

Software demos are not tours of your product. You don't need to take a deep dive into all your settings and obscure integrations, no matter how proud you are of them. Instead, think of a sales demo less as a presentation and more as a conversation. You should be asking questions and listening more than talking.

For more on sales demos, I recommend the book *Product Demos That Sell: How to Deliver Winning SaaS Demos* by Steli Efti.

A Hack to Lower Sales Effort

Sales demos are pretty high-touch, which means they should be reserved for customers who are going to pay you enough to be worth it.

You need a good process for qualifying prospects before they get to you so you're not stuck doing demos with people who will pay you $30 a month or are the wrong fit for your product.

Dialing in your positioning, website, and marketing is one way to make sure you're attracting the right prospects and weeding out those who aren't a good match. Using a qualifying form to schedule a demo is also good. Have them put in the company's name, the company's size, their best work email, and other information you need to know.

Weeding through those prospects can be time-consuming—especially if you have a dual funnel with low-priced and enterprise-level tiers.

Here's a hack: At Drip, anytime someone clicked "Book a Demo," they got a pop-up that asked for their name and value metric (i.e., how many subscribers they had). If they put in a low number, they were redirected to a page with a video demo, a 10-minute screencast of me walking through the product. If they put in a high number, they were directed to our scheduling link to book a time for a more extensive conversation.

As Drip grew, the cutoff number for in-person demos grew, too. At first, we were doing demos for people in our lowest tiers because it was early and we wanted to learn about our market by talking to anyone we could. Bit by bit, we ratcheted up the number on the form based on how many salespeople had the bandwidth to run demos.

When Should I Hire Someone to Take over Sales?

Hiring someone to help you with sales comes down to whether you're good at it and whether you enjoy it. If you're good at sales, it makes sense to keep doing it until sales conversations are encroaching on the rest of your founder responsibilities.

If you're bootstrapping, you have to hustle in the early days until you have the money to hire someone. But by the time you have five or six employees, you probably should only be involved in large deals. Sales demos are usually easy to teach, and as a founder, your energy will be better spent elsewhere.

One thing to note is that if most of your leads are warm, inbound leads, you can actually combine the sales role with customer success.

This works best when your prospects already have some sense of your product and are just trying to understand whether or not it's a good fit for them. Your customer success/salesperson will be there to show them around and answer questions rather than lead high-pressure sales calls. If it's more cold/complex sales, you'll want a salesperson incentivized by commission.

Team

How Should I Structure My Team?

In the early days, you'll be handling support, writing code, running marketing experiments, doing sales demos, and onboarding customers. You wear all the hats, which means you cannot focus on one role.

If you want your business to grow, you have to start peeling off some of those hats and giving them to other people.

But a challenge most founders face is, because you perform three (or ten) roles at once, you think you can hire people who can also handle many roles. It's easy to dream of hiring someone who's good at customer support, business development, and front-end design when you're handling those tasks and none of them add up to a full-time position. But these three disciplines require specific skill sets unlikely to exist in a single person.

When building your team you should delegate roles, not tasks.

So how does one do that?

The good news is there aren't that many departments in a SaaS company. The following is a list of the departments you'll need (plus a likely first hire in each):

Product. Product is focused on what to build and how it should function. Your first hire in this area will likely be once you're above $1 million ARR, with a title like Product Manager.

Remember that "Manager" at the end of a job title implies they manage a process, not people. "Manager" at the beginning of the title implies they manage people.

Design. Design works with product and engineering to help define how new features should look while also weighing in on how they should function from a user's perspective. Your first hire here will likely be a UX Designer.

Engineering. Engineering is focused on writing code and managing production servers. Your first hire here will likely be some variation of a Software Engineer.

Marketing. Marketing is focused on generating and converting inbound leads. It usually includes marketing strategy (deciding which approaches to tackle), implementation, and business development.

Your first hire here will depend on your marketing funnel, but typically you start with individual contributors who have experience implementing one or two marketing approaches (e.g., SEO, content, PPC).

But eventually you will need to hand off marketing strategy and project management to a Manager/Director of Marketing.

Sales. Sales brings in qualified leads and closes deals. This role is divided into a Sales Development Representative (SDR) or Business Development Representative (BDR) who qualifies leads for an Account Executive (AE) to close the deals. Your first hire here will likely be an AE who qualifies and closes their own deals.

Customer Support. Support answers incoming support emails, chats, and phone calls. Your first hire here will likely be a Customer Support Representative.

Customer Success. Customer success focuses on getting customers to stick around; their goal is to help people get onboarded and not churn. Your first hire here will likely be a Customer Success Manager.

Human Resources. Human resources (HR; sometimes called People Ops) is focused on compliance, payroll, people ops, and organizational structure. Your first hire here will be a ways down the line but will likely be a combined Operations role that handles HR, Legal, and Finance. This role typically has a title like Operations Manager or Director of Operations.

Legal. Legal responsibilities are outsourced to an external lawyer, with the founder handling coordination until there's enough budget to hire someone in an Operations role.

Finance. Usually outsourced to a bookkeeper and accountant, with everything beyond that handled by the founder (for now) until there's enough budget to hire an Operations person.

What Role Should I Fill Next?

Being a founder means firing yourself from one job after another to focus on the high-level strategic roles the company needs you to handle.

Before you can fire yourself, you have to hire someone to fill that role. To determine which role to fill next, track your time for a week or two (or you can list from memory) to compile a list of all the tasks you're handling. As a founder, these will usually span many departments and many roles.

Then, using the list of departments I provided above, group your tasks into those departments (and if possible, individual roles).

Once you have that list, ask yourself:

- Which of these am I bad at and think someone else could do better?
- Which of these am I good at but don't enjoy?
- Which of these could I stop doing that wouldn't cause a negative impact?
- Which of these could I hand off to a current team member?
- Which of these, if leveled up, is most likely to grow the company?

In a perfect world, you'll find that what the company needs most urgently is one of the things you're not good at or don't enjoy. When that doesn't happen, you'll have to decide which role to prioritize. It seems there is never enough money to hire for every role you'd like to.

As a general rule: support is usually an early hire because it's a repetitive task of lower value than other things a founder is typically focused on.

Sales, marketing implementation, and development usually come next; which one depends on the skill sets of the founders. If you have two founders who are both engineers, you likely don't want to hire an engineer. In that case, you'll want to do some soul-searching to determine if one of you should dive deep into marketing or sales or hire for it. Hiring for a role a founder has never done can be quite challenging.

Can I Combine Roles?

When you have roles that don't require 40 hours a week, it can be enticing to want to combine two or three roles into a single hire. It's hard to imagine hiring a full-time person to handle something you only spend 10 hours per week handling.

One reminder I have for you is that, as a founder, you are often more effective than the average hire. So realize something that takes you 10 hours per week might take them 20.

Second, as the founder, you are often not doing the best job handling certain tasks because you're in a hurry as you task-switch from one to the next. Someone focused on a single role will do a more thorough job, requiring more time than you expect.

But unless you've raised a chunk of funding, when you only have 10 or 15 employees, you'll want to look for generalists who can fill two roles at once. Eventually, you will hire specialists with specific domain expertise who fill a single role.

With that said, certain roles combine more easily because they require similar skill sets. If you are going to combine two roles into a single hire, here are some typical combinations that work:

Customer Success + AE*

Customer Success + Customer Support

AE + BDR / SDR

* Only with warm, inbound leads

Product Manager + Designer

Engineer + Product Manager

Engineer + Designer

Marketing Strategy + Marketing Implementation

Marketing Strategy + Marketing Project Management

Legal + Finance + HR (Operations)

Finding someone who is good at and wants to handle multiple roles will be more difficult than finding someone to fill just one role. But in the early days, you won't have much of a choice.

As your team grows, you'll start splitting these roles using the same principle of firing yourself from roles you're not a good fit for. If your Customer Success Manager is great with your high-end customers, hire someone to take their support workload off their hands so they can focus on retaining your best customers.

When splitting roles, have your team member run through the task-tracking exercise described above to help determine which role you most need to hire for.

Don't Invent Job Titles

I used to make up job titles because, as a bootstrapper, I didn't particularly care what someone's title was. I didn't want it to matter—but it really does.

When we realized we needed an architect to scale our infrastructure at Drip, we asked our internal recruiter to hire for the job of "Senior Scaling Architect." She eventually talked us into the title of "Senior Architect." Why? Because when she ran the data, she couldn't find enough salary information on the title we'd given her. Not only that, but if we'd used a made-up job title, qualified candidates wouldn't have known what we were hiring for.

There are standard SaaS job titles. Use them. Your ideal candidates have saved job searches for things like "Engineer," "Customer Service Lead," and, yes, "Senior Architect." Ignoring that makes it harder to connect with people searching for the job you're hiring for. It also does a disservice to whomever you end up hiring. They'll have a much tougher time explaining their qualifications to their next employer when their job title was "Code Wizard" rather than "Senior Engineer."

Although a treatise on organizational structure is beyond the scope of this book, here's a typical hierarchy of engineering titles (in descending order of authority) that can be easily translated into other departments:

- Chief Technical Officer
- VP of Engineering
- Director of Engineering
- Manager of Engineering
- Senior Software Engineer
- Software Engineer
- Junior Software Engineer
- Entry-Level Software Engineer

Note: These titles assume the typical path is to move into management, which doesn't have to be the case. Individual contributor titles above Senior exist, such as Principal Engineer and Distinguished Engineer. But for the sake of simplicity, I'm laying out the above hierarchy, which will work for companies well into the millions of ARR.

Another note on titles: be careful with handing out elevated job titles to early employees. One company I know named their first customer service person "Head of Customer Success." When they inevitably grew and added more customer service people, they didn't want him managing them and ended up in a tough situation. Should they demote him and have him leave? Or come up with an even more elevated title for the real manager?

A Note for Technical Founders

SaaS founders generally come in three flavors: those with a software development background, those with a marketing or sales background, and those who are subject matter experts.

I just got done recommending you hire for the roles you don't enjoy or aren't good at. This is especially true if you're a founder who's great at sales or marketing. It's a no-brainer to hire more technical help to free up your time to land more customers.

However, if you're a founder who's really good at the technical side of things, my advice is different. I recommend you focus on building skills in sales and marketing and start hiring developers to get yourself out of the nuts and bolts of the code.

There are two reasons for this.

First, developers are gonna develop. If your sole focus is on the product, you will want to solve every business problem by developing. Is your revenue plateauing? Are your churn numbers high? Are you not closing sales? The solution is to code more features . . . right?

Actually, building more features probably isn't the answer to those problems. But if writing code is comfortable for you and marketing is not, you'll have a natural tendency to over-build the product instead of getting to the real source of the problem.

Second, coding is deep work. You need to get into the Zen state of flow to have the right headspace. But looking up at the clock and saying, "Oh wow, it's already 4 o'clock!" is antithetical to handling all the day-to-day marketing and sales tasks you need to do as a founder.

As a developer, you need to be on what Paul Graham calls a Maker's Schedule, where you can take a whole afternoon to immerse yourself in work uninterrupted.

As a founder, you will absolutely be on a Manager's Schedule, which cuts the day into one-hour increments you can fill with all those necessary meetings and other tasks.

Of course, you can ignore this advice. You're in control; it's your company. Just know that you will hamper growth if you keep the job of developer forever.

A Note for Nontechnical Founding Teams

Ninety percent of bootstrapped SaaS companies have at least one technical founder, according to MicroConf's State of Independent SaaS Survey. If you do not have a developer on your founding team, you are very much in the minority.

> 9 in 10 bootstrapped SaaS companies have one or more developers on their founding team.

There's a reason for this: building SaaS is both complex and expensive. Aside from marketing, sales, and support, you have to write code and maintain an always-on production environment. Developers who are good at SaaS are usually not cheap, and being able to tell the difference between someone who says they are good and someone who is *actually* good is close to impossible without knowing how to evaluate their code.

Having no developers on your founding team makes SaaS difficult to bootstrap. Most of the founders I see who attempt this either raise funding very early or run another business that throws off enough cash to allow them to hire a senior developer out of the gate. Whereas, if one of the founders was a developer, they would build the product during nights and weekends for no out-of-pocket cost.

If you have no technical founders and want to launch a SaaS, find a developer you can trust early and expect to pay them a lot. You will almost need to think of them as a cofounder because they will be making technical decisions that will have a major impact on your company's future. If you don't have the budget to do this, I encourage you to find a developer cofounder.

Craig Hewitt, the nontechnical founder of Castos, told me, “*If I did it again, I’d very much want my first developer to be a cofounder or someone I know really, really well. I’ve wasted too much money while being misled by developers who are just looking to make a few bucks and don’t care about the outcome of the project.*”

What If I Don’t Plan to Hire?

If you’re a lifestyle bootstrapper who’s happy without scaling, you may not be planning to hire. And that’s great, but I’d still challenge you to consider hiring at least a support person.

Why? Because for bootstrappers, a lack of money isn’t what kills businesses. It’s founder burnout.

Sure, you’re good at answering support emails—you know the product inside and out. Sure, it might not take much time out of your day. (Or does it? I bet it takes more time than you think.) But that doesn’t mean you’re the right person for this job.

One objection I hear repeatedly is, “My product is so technical, I’ll never be able to find a support person who can handle it.”

This is rarely true. It’s not that your product is too technical; it’s that you need to take the time to set up documentation, systems, and training. I’ve seen super technical products where people found a sharp junior developer who could provide amazing support. Eventually that junior developer will graduate to an actual developer in the company, and you can replace them.

In the long term, handling support will likely lead to burnout, and hiring a frontline support agent will free you up to keep doing the work you love.

Be a Team, Not a Family

You’ll hear a lot of people talking about how their company is all one big happy family, but I caution against that.

I do not refer to my fellow employees as family. I view us as a high-performing team (this sentiment was popularized by Netflix in an early culture document).

Calling your employees your family is disingenuous. It may give you warm fuzzies to say it, but you don't fire your sister or uncle. You do bench a teammate if they're no longer the best player for their position.

The team mindset allows healthy interpersonal relationships to develop and friendships to be built without putting your business at risk. Teams that are "families" become enmeshed and don't maintain appropriate boundaries.

When you hire, be up front about your company culture: you're a team, you support each other, and you work together for the success of the business. If a teammate isn't performing, you need to be able to make the right decision for the company.

High-performing team members should be happy to hear it. We've all worked with people who aren't as good as the rest and who, frankly, drag the rest of the team down. It sucks to constantly pick up the slack for someone who's always dropping the ball, doesn't get things done as fast as you do, or doesn't care as much as you do.

If you hire and tolerate mediocre performance, you will lose your best people and create a culture of underperforming.

No one has ever said, "I fired that person too soon." Normally the regret is that you waited too long to fire someone and they dragged morale down around them.

Hiring Managers ×

I separate management into two components: supervision and leadership.

- A supervisor is someone who approves vacation time, gives reviews, and decides on salary increases. They basically handle the nuts and bolts.
- A leader—like a tech lead or a development lead—is someone who gives technical guidance, mentorship, seniority, and direction.

A manager is a person who supervises *and* leads. In a lot of companies, especially in the early days, that role is divided.

At Drip, my cofounder Derrick led our team when it came to technical guidance and code review, but the developers reported to me.

Derrick didn't need to get bogged down with scheduling vacation time or health benefits—that would have been a waste of his skill set, and I didn't want him burning cycles learning management when there were features to build. So those supervisory decisions came to me.

You can—and should—hire or promote roles like development lead or customer success lead early to keep every decision from falling back on you. Even before you have the budget to hire a full-blown manager, you will likely find stand-out folks you trust to make decisions for a particular project or department. If you don't find leads early, you will eventually find yourself overwhelmed with the volume of questions that come back to you because you haven't put the proper expertise in place.

If everyone who works for you is a task-level thinker (see sidebar), it places a tremendous burden on you as the founder. As you grow and transform from a product into a business, usually in the $20,000 to $40,000 MRR range, it's time to consider bringing some project-level or owner-level thinkers onto your team.

Task-level thinkers are team members who focus on their current or next task. They might be early in their career or get overwhelmed with more than a few sequential tasks on their plate. Most of us begin our careers as task-level thinkers because prioritizing many complex, interrelated tasks is often not a natural ability.

Project-level thinkers look ahead weeks or months and juggle multiple priorities. They often rely on team members to complete work that's combined into a single deliverable. Project-level thinkers have advanced systems in place to track the myriad moving parts needed to successfully complete a project.

Owner-level thinkers not only manage projects but also think about how to improve internal processes and bring ideas for experiments that can change the trajectory of the company. Owner-level thinkers look ahead months or years and consider strategic shifts that may need to take place to take the company to the next level.

Not only will this remove stress from your life, but it will also help your team move faster because they don't need to wait on you for every decision. And if you plan to sell the company at some point, the less the business relies on you, the better terms you're likely to receive.

When Should I Hire Managers?

When you're just starting out, you won't have the budget to hire a manager. It depends on several factors, but I usually recommend finding leads once you have two or three people in the same department and finding a manager once you have four or five people in the same department.

You can stretch this, but it will take a toll on you. It takes up too much of your time if you're managing twelve direct reports well (e.g., holding weekly or monthly one-on-one meetings, working on career development).

Usually, founders let it go for too long and then start managing their team poorly because they are pulled in too many directions. This can work for a while, but it will eventually lead to poor results or folks leaving your company.

Keep in mind that your first management hires must also be individual contributors. You'll want to find someone who can manage *and* develop. Or manage *and* do customer support or write documentation or work on strategy.

Promoting managers from within can be a good way to go if you have someone ready to move from task-level to project-level. In this case, you take that promising tech lead or head of customer success and train them to run one-on-ones, evaluate for raises, approve vacation time, and—most importantly—give both positive and negative feedback.

What Makes a Good Manager?

Just because someone is a good lead doesn't mean they will be a good manager. Not everyone has supervisory skills or the desire to learn them. If you hired your most senior developer because of their ninja coding skills, it doesn't make sense to bog them down managing time-off requests and answering questions about health insurance.

I find that people who've had a good manager in the past tend to have a leg up when it comes to being a good manager themselves. The problem is that many bootstrapped founders have never had a "real job" inside a bigger company. They haven't experienced a range of managers or aren't familiar with office politics, so they don't develop a mental model of managing a team well.

One of the more challenging things about becoming a manager is learning to give negative (constructive) feedback. Most people are nice, and most of us don't like conflict. It's not fun to have to tell someone they dropped the ball—but it is a necessary skill to learn to help your team members grow and keep the company culture from slipping into mediocrity.

My rule is to praise in public and correct in private. If I'm going to tell someone they screwed up, it'll be in a one-on-one meeting and never in front of someone who wasn't involved. Team meetings are for giving out props to team members who are doing a great job, not tearing someone down for a mistake.

Giving both positive and negative feedback builds trust, which is the most important part of being a good manager.

Your team members need to trust that you'll be fair. That you'll listen to them. That you'll make good decisions. That you'll take the fall instead of throwing them under the bus when things go wrong—because if someone below you makes a mistake, that ultimately falls on you.

They also need to trust that you'll tell them the truth, even if the news is not good.

One company I worked at had a problem where the managers were too nice. Eventually when they had to let someone go, that person was shocked because no one had ever told them something was wrong. Of course, they then spoke with old coworkers via a back channel, word spread around the company, and suddenly everyone was wondering if they were next. The trust was broken.

How Can I Hire Great People?

Now that you know the role you want to fill, how can you find the best candidate?

I'm a staunch supporter of taking your time to fill a position. The people you hire—particularly at this phase of your business—can either help you grow or hold you back. It's better to be picky and learn to hire slow and fire fast.

After spending hundreds of hours hiring at my last startup, I swore I would never hire on my own again. If you have the money, hire a recruiter to help you find the best candidates.

Many recruiters will work off commission, such as asking for 15% of the first year's salary. Unfortunately, they are more expensive than the value they bring. I recommend going with a flat-fee recruiter like Remote First Recruiting.

Whether or not you have the money to hire a recruiter, these tips will help you find and attract the best candidates.

Be Different. When I was hiring for Drip, I would write job descriptions explaining that we viewed software as a craft. That we often wrote 2.5 lines of test code for every line of production code and that our codebase was immaculate.

I wanted to attract developers who viewed themselves as craftspeople, who were looking to work in an environment where that meticulous dedication would be valued.

When writing job descriptions, you want to show folks that working at your company will be unlike other places they've worked. It's not enough to say that you're different; you have to *be* different.

Work on Interesting Problems. Amazing team members want to be challenged and learn something new each day. Your job description should lean into this.

It's helpful if you happen to be working on an exciting, innovative product, but it doesn't need to be. Building an email services provider like Drip wasn't inherently interesting. But when I would recruit developers, I would engage them with the things that were interesting.

- We were scaling fast, which is an interesting problem.
- We had a wide variety of UX paradigms, so the product itself looked great and was fun to work on.
- There was so much going on in the app that a developer's day-to-day job would be varied.

Lean into Your Advantages. As a bootstrapped or mostly bootstrapped company, high pay and benefits probably aren't going to be your advantage. Those are the advantages of Fortune 1000 companies. But those companies also have Achilles heels: office politics, no remote work options, massive teams, tons of meetings, and other headaches that many great team members don't want to deal with.

Your advantage is in finding what motivates these top performers beyond pay and benefits and giving it to them.

For example, here's a pitch I could give to any role I was hiring for: "We're fully remote. We're a small team with no politics. You'll work with and learn directly from the founders. We're a fast-growing company. You'll work on interesting problems."

For specific roles, I added even more advantages.

I told developers, "You're going to have a huge impact on the product and the company. You'll work on an amazing code base. We ship features weekly, so you'll always be working on something new."

I told customer success folks, "We have a great product, and our customers love us, which makes your job easy. We also have a high feature velocity to make sure it continues to be easy."

Think of Your Job Description as a Sales Letter. Your job description should convince the person reading it that you have a fantastic company and that they should apply.

You're not a big company, so don't write like one. One of your advantages is that you can inject personality into your job descriptions, even to the point of making them entertaining. Write like a human, and don't shy away from having personality.

At Drip, I had a line in our developer job description that read, "Crappy code makes you want to flip a table." It's a quirky thing to say and made people realize our team had personality and opinions. It was an instant filter to drive away people with whom that thought wouldn't resonate and attract those with whom it did.

When you think along the lines of a long-form sales letter rather than a list of requirements, it will set you apart from the stiff Fortune 1000 job descriptions and communicate something about your company's culture by showing, not telling.

You should also let candidates know you're picky about who you hire. A-level performers want to work with other A-level performers, and they get excited by the idea of joining a top-notch team.

Build an Audience. I struggle a bit with this advice because you can build an amazing company without an audience. I've seen hundreds—thousands—of founders who have built strong companies without an audience.

I also don't believe you need to build an audience to build a successful SaaS product (in fact, I think it's a waste of time for most people).

That said, having an audience makes hiring easier. Because when you have an audience that knows you, likes you, and trusts you, it's much easier to attract good talent.

Top-tier talent is constantly learning. They are reading blogs and books and listening to podcasts. If you are one of the people they learn from, you'll find a higher percentage of top-tier talent in your audience when it comes time to hire.

Remember: Hiring Is Hard, but Retention Is Critical

I want to point out that all the hiring tips in the world won't help you if you're not focused on retaining your people. If you misrepresented what the experience would be like, or you don't keep them interested in the work, top performers will move on because they have the option to work anywhere.

For further reading, I recommend the book *Who: A Method for Hiring* by Geoff Smart and Randy Street. It gives a solid framework for the entire hiring process. In addition, it provides a solid interview approach to discover more about what candidates actually did in their previous roles and teaches you how to avoid asking surface-level questions.

For further reading on job descriptions, I've seen a few companies emulate Drift's approach of creating a 90-day plan for every new role, and that effectively becomes the job description. You can read about Drift's hiring process here: drift.com/blog/hiring-philosophy/.

Where to Find Qualified Candidates

If I were hiring right now, there are a handful of websites I would use to advertise the job listing. Because these sites change over time, I have a page on this book's companion website that includes the most up-to-date hiring sites for SaaS founders: saasplaybook.com.

The primary benefit of these job boards is that candidates are typically looking for roles at remote or small startups.

Should I Offer Equity, Stock Options, or Profit Sharing?

Bootstrapped founders face a big question: How do I incentivize employees beyond the agreed-upon salary? —Bonuses? Profit sharing? Stock? Equity? First, let's talk about why this might be a good idea.

Hopefully your employees love working for you because of the interesting things they're working on. Extra incentives—when properly aligned with the company's goals—can provide motivation and retain employees. And in a competitive talent market, smart incentives can help you land top candidates.

It's not a requirement to offer bonuses or profit sharing, but if your team is cohesive and working hard toward the same end goal, you're creating profit, value, and wealth together. It feels right to share that with your team.

(Note: Nothing in this book should be considered legal or tax advice. Speak with a lawyer and accountant before starting an employee incentive plan.)

Bonuses

Bonuses sound great because of their flexibility, but they are tricky, given their arbitrary nature. They are a decision to give people an extra few thousand dollars at the end of a year.

Bonuses can make people feel left out or that you're playing favorites. They might feel like you're giving more money to someone who doesn't deserve it.

If you don't have a profitable year and don't give out bonuses, people can get angry and blame you. They'll point out how much money you spent on things they don't like. Not to mention in California, a lawsuit ruled in favor of employees over nonpayment of bonuses those employees had come to depend on.

Usually, it's better to incentivize employees with something more aligned with the business goals you're trying to achieve.

Equity

Equity gives employees ownership—and not just literal ownership. It gives them emotional ownership of the business and motivates them to grow it.

Equity is tricky, though. An equity holder only makes money if you sell the business or pull out dividends.

With a venture-backed startup, selling is often the goal, and equity comes with the promise of large liquidity in the relatively near future. Because bootstrapped startups tend to grow more slowly and deliberately, equity isn't always as much of an incentive.

Another challenge of equity grants is that they're different from stock options. You're literally giving a portion of the company to someone, which is taxable on the current value of the company. You might create a brutal taxable event for an employee if you give them a substantial amount of equity (or even an insubstantial amount if the company is large enough).

In addition, if you're a pass-through entity (such as an LLC in the US), capital gains will pass through to any equity holder.

Let's say your LLC makes $500,000 in profit this year, and you've given 1% equity to a key employee. Even if you haven't pulled money out, they will receive a K-1 for 1% of that $500,000, or $5,000. Essentially, they're getting taxed on $5,000 even though they didn't receive that money.

In most cases, equity is best for founding employees (cofounders) who receive it when it's virtually worthless and they know the ramifications.

Stock Options

Stock options are the standard startup approach to getting deeper buy-in from employees. An option just means an employee has the option to purchase a share of stock in the company.

If you grant someone 10,000 options, they can purchase 10,000 shares at a fixed price (called the strike price) set each year by a company's IRS filing. That strike price is usually quite a bit less than the share valuation during the last funding round, which means it's a good deal, at least on paper.

Those 10,000 options vest over time, the standard is four years, creating an incentive for the person to stick around so they don't lose their unvested options.

From the company's side, you would set up an options pool of 10% or 15% of outstanding shares that are given out in small chunks to new hires, the amount varying based on seniority.

Stock options have simpler tax implications than equity because they are a promise to the employee that they can buy shares at a future time rather than actual shares.

Stock options are a reasonable choice for employee incentives, especially if your goal is growth and an exit rather than running the company for the long term.

Profit Sharing

The nice thing about profit sharing is that it doesn't require you to sell your business for your employees to make money. If your goal is to make your company profitable and run it for the long term, profit sharing may be your best option.

Consider structuring profit sharing as a pool rather than a committed percentage to an individual. For example, instead of telling early employees that they'll get 1%, 2%, or 3% of profits, have all key employees share in a 10% or 15% profit-sharing pool.

As you add more people to the pool, those first employees' percentage of the pool will go down. But ideally, profits should be growing, and every team member should be contributing to that.

Peldi Guilizzoni, founder and CEO of Balsamiq, wrote perhaps the best explanation of profit sharing I've seen. To read it fully, go to bit.ly/balsamiq-profit-sharing. Essentially, the company started with a pool of 10% of the profits, which was distributed each quarter. At some point, years into the company, he increased that pool to 15%. I believe he's now up to 20%.

That profit pool is allocated to full-time employees. Twenty-five percent is split equally, and 75% is based on seniority, then it's weighed by the cost of living for each employee.

Guilizzoni notes that they do quarterly distributions because monthly was too much paperwork and yearly kept some unhappy people around longer than they should have stayed.

Some companies have folks vest into profit sharing for their first few months, much like some companies have a waiting period to get health insurance or to access a 401(k). This is a way to make sure the person's a fit for the team and that the team is a fit for the person.

Of course, if your plan is to grow and exit rather than run profitably, profit sharing is likely not your best option.

Do I Need a Cofounder?

Venture capitalists tend to look unfavorably on single founders, partly due to the opinion of Y Combinator founder Paul Graham. In the bootstrapped SaaS space, though, single founders make up many of the most successful companies. In fact, over half of the respondents (56%) to our 2022 State of Independent SaaS Report were single founders.

The short answer is: no, you don't need a cofounder. In fact, the most equity you will ever give away is to your cofounder. No investor, employee, or advisor will come close to owning the amount of equity you give away when starting a company with another founder and splitting it 50/50.

Being a single founder has its own list of pros and cons. The beauty of being the sole founder is the simplicity. You don't have to play nice with a cofounder, and you get to make all the decisions.

One of the major downsides is that you need to make all the decisions and do all the work yourself. Especially in the early days when resources are tight, this can be isolating and mentally challenging.

Most of the single founders I know have a strong network, whether it's other founder friends they can lean on when things get tough, mentors or advisors they can bounce ideas off, or a long-term mastermind group (I talk more about masterminds later in the book) that offers ongoing support and advice and serves as a sounding board.

There can be enormous value in having a partner in the trenches with you as you travel this long journey. Here are some thoughts to keep in mind if you're considering a cofounder:

Are Your Skills Complementary? Too much overlap means you'll argue over certain areas of the business and neglect others. The best mix I see is a developer founder pairing with a marketing/sales founder or a developer and a subject-matter expert.

How Well Do You Know This Person? You are effectively entering a marriage. If you don't know the person well or have never worked with them before, take things slow. "Date" by working on small projects before equity changes hands.

Are You Protected If Things Don't Go Well? Talk to a lawyer. Make everyone's equity vest, typically over four years. One of the worst things you can do to your company is let a founder walk away with 50% of it after working on it for a few months. At my SaaS accelerator, TinySeed, we wanted to but sadly declined to fund several companies in this situation.

How Much Value Does a Cofounder Add? The biggest question to ask yourself is: Will joining forces with this person make the company more valuable? Will it help it grow faster? Although this is a difficult question, I've found that being on the fence about whether someone will bring value means you should probably find another option.

Too Many Founders

Some companies I see don't need more founders. Sometimes they need fewer.

The partnership between cofounders is almost like a marriage. There are interpersonal dynamics and intense decision-making. There's financial responsibility. At times you'll spend more time together than with your significant other.

Add a third or fourth person into that mix, and there can easily be too many opinions to serve the business well.

I understand the impulse. Starting a business as a solo entrepreneur is scary. Starting one with a partner is less scary. So why not bring a few more onto the bus?

What often happens is that when three or four friends get together to start a company, one of them is a weak link in the chain. If you're in this situation, you may already know in your gut who isn't pulling their weight. And that weak link will be detrimental to the business—both in terms of the increased growth you need to justify having that founder on board and because there are too many chefs in the kitchen.

80/20 SaaS Metrics

Management guru Peter Drucker said, "*If you can't measure it, you can't manage it.*"

Most founders get overwhelmed when they think about metrics because they don't know which to track or are tracking too many. This section will help you narrow your focus to the core SaaS metrics that matter most (the 20% of metrics that will drive 80% of your results).

Which Metrics Should I Track?

As a founder, data is your copilot. You should know your most important numbers if you're going to grow your business.

You don't have to be a metrics savant, though. I've seen founders build relatively successful businesses without getting mired in the numbers (though often in spite of that fact, not because of it). I've also seen many companies take longer to gain traction because their founders don't track basic metrics—or put too much emphasis on the wrong ones.

Two "North Star" Metrics

Your two most important metrics are MRR and month-over-month growth rate. Those are the top-line numbers that indicate how far you've traveled and how fast you're going. Every SaaS founder should look at these numbers on at least a weekly basis.

But MRR and growth rate are lagging indicators.

To that end, I'm going to give you a metrics framework called 3 High/3 Low to help you zero in on the other important key performance indicators of your business.

(Note: These are overall business metrics. Separately, there are marketing funnel metrics you should track, which we covered in the chapter on marketing.)

Tracking the 3 High/3 Low Metrics (six in total) will tell you two important things:

1. How healthy your business is
2. When your revenue is going to plateau

These metrics aren't difficult to track. A dashboard like ProfitWell, ChartMogul, or Baremetrics can connect to your payment processor and give you the data at a glance.

You'll notice that many of these six are in tension with each other. You want the Low Metrics to be as low as possible and the High Metrics as high as possible, but often when one is going down, it's causing another to increase.

I'll provide some rules of thumb in the following section, but keep in mind that your product, ideal customer, and industry will have a substantial impact on these numbers.

3 High/3 Low Metrics Framework

The 3 High/3 Low framework includes the next six metrics you should be tracking after MRR and growth, and you want to push three of them upward (i.e., high) and three of them downward (i.e., low).

We'll start with the three Low Metrics.

LOW: Cost to Acquire a Customer (CAC)

In its simplest form, CAC is all the costs associated with landing new customers (e.g., marketing, advertising, sales) divided by the number of customers you acquired during that period.

It's sometimes tricky to calculate because getting a handle on your marketing costs can be tricky. If you're focusing on SEO, you may be creating all the content yourself rather than paying a writer. You may be getting a lot of your early customers from forums you spend time on or by getting in front of other people's audiences. In those cases, the cost is your time rather than an easy-to-calculate number.

It's a lot simpler to calculate CAC if you're running ads. Then, you can see how much you're paying per click and track how many people convert from each source.

But if you're not in that position, valuing your time at a certain rate (e.g., $150 an hour) and taking your best guess at time and money spent on marketing in a given month can get you to a good enough estimate of your CAC.

How do you know if your CAC is too high? By calculating how long it'll take to pay back the costs of acquiring each customer.

As I was first getting into recurring revenue, I thought that if I was getting $1,000 in LTV from each customer, I could spend $700 to acquire every customer and make $300 a pop. Right?

The problem is that you're not getting $1,000 every time you sign a new customer. With a $50-a-month contract, you're getting that $1,000 over the course of the next year and a half.

If you spend $700 per new customer in January, you won't break even on those customer acquisition costs until next February (assuming the customer doesn't churn).

With venture capital, the rule of thumb is that you should spend no more than one-third of your customer's LTV or no more than one ACV.

As bootstrappers, we don't have enough cash to wait 12 months to recoup CAC from every customer. Most successful bootstrappers I know are in the two- to six-month payback period (depending on how much cash they have in the bank).

There are times when that number can get more aggressive. For example, at our peak with Drip, we could afford to spend more on customer acquisition because we had the cash in the bank and I knew the numbers in the rest of our funnel by heart. Even at our peak, though, we were only running seven or eight months out—that's the high end for bootstrapped companies.

LOW: Sales Effort

Sales effort is a measure of the length of your sales cycle and includes the number of touch points required to make the sale. Where CAC measures the amount of money you're spending to get a new customer, sales effort measures the time and energy you're spending.

The best way to track sales effort is to look at both the average number of days from someone scheduling their first demo to closing and the number of calls it takes to close a deal.

Your ability to keep sales effort low depends greatly on your industry and customer base.

If you're doing enterprise sales, your sales cycle will be long and require more effort than if you're targeting solopreneurs and other small businesses with a single decision-maker. A three- or four-month sales cycle is reasonable in enterprise sales—and worth it because the ACV might be $50,000. If you're spending that much time for $5,000 contracts, though, that's rough.

No matter what your sales process looks like, you want your sales effort to be as low as possible. Here are some ways to lower this number.

Self-Serve Sign-up and Onboarding. Many inexpensive products can get away with low price points because they have a low-touch or no-touch sales process. They have a self-serve sign-up and onboarding process, which requires almost no sales effort.

The higher your ARPA, the less likely they are to become customers without some sales effort. But finding places to offer self-service along the journey can reduce the amount of hand-holding your team has to do while making the process speedier for your customer.

One-Call Close. Self-service isn't going to work in a lot of spaces, but you can try to get to a point where the decision is made by a single person. You can do this by targeting a founder, a developer, or a single manager.

You can also streamline the back-and-forth of providing more sales materials, getting on second calls, waiting for input from the committee—and on and on.

Educate your customers as much as you can ahead of time so they have the information they need and develop checklists to gather the information you need to close the deal quickly.

LOW: Churn

Churn is the percentage of people canceling their subscription each month, and it's the Achilles heel that kills (or plateaus) SaaS apps. If you can keep churn low, growth is much easier. If churn is high, it's a force that's very hard to outrun.

Focus on revenue churn. To calculate this, divide the gross MRR that canceled in a given month by the starting MRR for that month:

$$\text{Gross revenue churn} = \frac{\text{Canceled MRR}}{\text{MRR at the start of the month}}$$

As a general rule, for most bootstrapped B2B SaaS businesses:

→ Gross churn > 10% = **Catastrophic**
→ Gross churn 8–10% = **Not Good**
→ Gross churn 6–7% = **Meh**
→ Gross churn 4–5% = **Fine**
→ Gross churn 2–3% = **Good**
→ Gross churn < 2% = **Great**

With this caveat: if you are focused on high-priced contracts, say, above $25,000, your churn should be lower than the chart above. In that case, I'd categorize fine churn as 2–3%, good churn as 1–2%, and great churn at or below 1%.

Churn is such a critical metric because it helps you calculate when revenue will plateau.

At some point, the number of new customers you acquire will equal the number of customers you churn out each month. This causes your growth rate to effectively hit zero. You've hit your maximum number of customers (and revenue) that you can achieve without changing something in the business.

It's a simple calculation:

$$\frac{\text{MRR added each month}}{\text{\% Churn rate}} = \text{Maximum MRR}$$

If you acquire $5,000 in new MRR each month and have a churn rate of 10%, that's 5,000/0.10 = $50,000. If you don't change something in your business, you will plateau at $50,000 in MRR.

This plateau number should strike fear in your heart because SaaS plateaus are brutal. They are often difficult to fix, as they might require a strategic overhaul of your product, customer focus, or marketing approaches.

I recommend every SaaS founder calculate their plateau number. You could feasibly have a dashboard that calculates it in real time. It gives you a window into the future and lets you start troubleshooting early to push that plateau further out.

Churn is such a critical metric that we'll dive deeper into it in the next section.

HIGH: Annual Contract Value (ACV)

ACV is the amount a SaaS customer will pay if they stick around for a year, whether you offer a yearly plan or calculate it based on 12 months of your monthly subscription cost.

A lot of SaaS resources will point you to tracking LTV, but ACV is actually the more valuable metric for SaaS bootstrappers. Here's why.

The simplest equation for LTV is your ARPA divided by your churn.

For example, if you're getting $50 a month from a customer and have 5% churn, your average LTV for each customer is $1,000.

This is far from a perfect formula, but it's the simplest one to get insight into your LTV.

Let's say you lower that churn to 1%, which makes your LTV $5,000. Pretty good, right? Except that you'll be getting that five grand over the next eight years. If you have millions in venture capital in the bank, maybe you can afford to wait a while to recoup your costs, but as a bootstrapper, you need to be thinking shorter term.

That's why I recommend you stay focused on ACV as a key metric rather than LTV.

One of the biggest ways to keep your ACV high is to sell to businesses rather than consumers—usually the larger the business, the more they can pay (though that depends on the problem your product solves). This metric in particular is usually in tension with CAC and sales effort because selling to more significant customers requires more sales effort, which is naturally more expensive.

You can also increase ACV by raising prices, which we covered in-depth in the Pricing chapter.

HIGH: Expansion Revenue

We discussed this in the Pricing chapter, but as a refresher, expansion revenue is when customers pay you more as they get more value from your product. Whether by manually upgrading to the next tier or being auto-upgraded as their usage increases.

You can increase expansion revenue by having pricing tiers that ensure the more value a customer gets out of your product, the more they pay. That can be through value metrics (like adding seats in a CRM or subscribers in an email services provider), feature gating, or both.

When your expansion revenue is high enough, you can actually get to the point where your revenue is growing even when you're not adding any new customers. That's what makes expansion revenue an incredible SaaS Cheat Code.

HIGH: Referrals

The last critical SaaS metric is referrals, or how many new customers were referred by your existing ones. Referrals are a good metric to monitor because it is less of a lagging indicator than most.

When your referral per customer is high, the natural flywheel of virality starts to spin. Over time, word of mouth can become a big driver of new customers—and certainly one of your highest-converting drivers. Referred customers have substantial conversion rates and take a lot less sales effort because they're already inclined to trust your product.

The best way I've seen for determining how many referrals you're receiving is to ask how a customer heard about you at their point of sign-up.

Asking for Referrals. Not every product can have word of mouth baked into the product, but every founder can—and should—be proactive about asking for referrals.

When you see that trials are converting well and customers are happy with your product, set up an automated email that goes out around the 60- or 90-day mark. Say something like, "So much of our business is based on referrals. If you're enjoying our product, could you please pass the word along?"

The automated email works well when you have a pretty hands-off, low-touch sales process. However, for products with higher ACVs and a more intensive sales process, it's better to ask for referrals in person.

SaaS Cheat Code: Virality

When we talk about a SaaS product having virality, it means that every user you add has a viral coefficient greater than zero. You can count on each new customer bringing at least a fraction of a new customer within a particular time frame.

Some ways of achieving virality are stronger than others.

Strong Viral Loop. When the value of your app is in how it connects people, you have a strong viral loop. Take Facebook, where people who join tend to invite their friends onto the platform. Slack is the same way. It's not useful if you're on your own, so when someone adopts the tool, they also generally invite the rest of their team.

In the above examples, people need to sign up for an account to use the product. But you can have strong viral loops even without that requirement.

When I send out a SavvyCal link to someone who wants to book time on my calendar, they click through the link and experience for themselves how easy the tool is to use. The fact that the link came from someone they know is a tacit endorsement. When they want to try a scheduling app themselves, they're already inclined to trust SavvyCal.

Electronic signature apps have that same built-in virality. When you send your SignWell link for a colleague to sign a document electronically, they are exposed to the experience of using it and may be inclined to choose it next time they have a document that needs a signature.

Weak Viral Loops. Making your brand visible on customer-facing interfaces is another form of virality.

When Drip customers on certain low-price plans put an email sign-up widget on their website, anyone who sees the pop-up also sees the words “Powered by Drip.” If they’re in the market for an email widget for their own website, they might check out Drip.

A lot of SaaS tools require branding or watermarking as part of their free plan. For example, Mailchimp adds a badge to the bottom of every email sent through their free plan. Veed.io, a video editing tool, adds a watermark to videos made with their free plan.

The first reason these are weak viral loops is that, even though they get your brand in front of potential users, it doesn’t create as strong a bond as when someone interacts directly with your product.

The second reason is that if someone is interacting with your product because they were invited (like in the Slack example), they’re probably closer to the target audience than someone who’s just on a mailing list or watching an influencer’s video on YouTube.

Building Virality. Although this SaaS Cheat Code can be incredible for growth, it’s extremely difficult to retrofit virality into your product. If you want to build a viral product, start from the beginning and actively pursue the business ideas that lend themselves to strong viral loops.

The good news is that you can absolutely build a SaaS company to millions in revenue with a zero viral coefficient. Virality is amazing when you have it, but it’s not a deal breaker for building your company.

A Quick Note on Vanity Metrics

Sometimes you’ll hear a founder bragging about getting 50,000 unique visitors a month to their website, getting 100 new free users a week, or having 25,000 people on their mailing list.

Those numbers sound impressive until you dig deeper.

How many of those unique visitors convert to trials? How many of those trials convert to customers? What's the open rate on that mailing list?

Vanity metrics like page views, subscribers, or free users are interesting, but only in context. The real question is: how many of those visits or free users are turning into paying customers?

How Much Should I Worry about Churn? ↘

Churn is the death of SaaS. I've seen multimillion-dollar acquisitions fall apart because of high churn.

In your company's early days, the actual number of churning customers doesn't matter much because your numbers are too small to be meaningful. If you have $1,000 in MRR and one customer paying you $200 cancels, that's 20% churn. Normally, I would consider that business-on-fire level—but in this case, it's just a single customer.

When you see a company with high churn, it's a sign something's not working. The product isn't where it needs to be, they're solving a problem no one needs solved, or they're getting the wrong customers through the door. At that point, they should be trying to refine product-market fit.

Until you have product-market fit, you should worry much more about *why* people are churning than the actual churn number. Of course, low churn is always better. But throwing churn-reducing tactics at your customers when you haven't built something people want and are willing to pay for can mask your lack of product-market fit.

There are many ways to game churn, like making people email you to cancel or moving to annual only, but those often hide the real reason people are churning, especially early on. The approach at this stage is to find out why people are churning and use that knowledge to refine your product-market fit.

Once you have decent product-market fit, your churn rate becomes highly relevant because it's how you know when you're going to plateau, and it's part of your LTV calculation, which is one indication of the strength of your business. Reducing churn is a critical part of keeping your business strong.

Generally speaking, the lower your product's price point, the higher your churn. Some of the TinySeed companies that cater to hobbyists or very small businesses have churn in the 5% range, and while I'd love to see churn lower than that, it's okay because they are in massive markets and have a very low cost to acquire new customers.

For most companies like the type we're discussing in this book, I suggest shooting for gross revenue churn as low as possible, certainly under 3% per month. At the venture scale, successful companies have less than 1% gross churn.

Segmenting Churn

If I tell you a product has an average review of 2.5 stars on Amazon of over 1,000 reviews, it probably sounds mediocre.

But if I tell you that it has 500 five-star ratings and 500 one-star ratings, that's a different story. Half of the product's users love it, and half are the wrong audience.

It's the same with churn rate. Saying you have a gross churn rate of 8% doesn't give you the right information to work with. But once you start looking at the churn rate of specific customer segments, you'll get a clearer picture of what's going on.

I like to segment by three things: pricing tiers, marketing channels, and time.

Segmenting by Pricing Tier

One TinySeed company has two pricing tiers: $30 a month and $100 a month. As they worked on their churn, they broke it into segments and found that Segment A ($30 a month or less) had a net churn rate of 11%, which is obviously a big problem.

On the other hand, Segment B ($100 a month or more) had a net churn rate of -4%.

Negative churn? What does that even mean?

Net churn is when you subtract expansion revenue, which is when customers pay you more money when they get more value from your product—usually when an existing customer upgrades to a higher pricing tier.

Your business achieves net negative churn when your expansion revenue outpaces the revenue you lose from churning customers. It can be hard to wrap your head around, but +4% churn means you're losing 4% of your recurring revenue each month. And -4% churn means you're gaining 4% of additional recurring revenue each month without adding new customers.

(We cover net negative churn, one of the SaaS Cheat Codes, later in this chapter.)

So for this company, its high-paying customers who make up 80% of their revenue have negative churn. It's night and day trying to grow a company when you have 11% churn versus 4% net negative churn.

This is the value of segmenting churn instead of viewing it as a single number. It inevitably helps you understand your business better than if you were viewing churn in aggregate.

This company's numbers aren't unique. In almost every case, your low-paying customers will churn faster and your high-paying customers will be stickier.

Is the answer in this case to cut your lowest pricing tier? It depends. If your lower tier allows people to try out your product and you're seeing a decent amount of conversion from that tier to a higher one, it might be worth dealing with the high churn rate. Especially if the lower tier doesn't require a lot of expensive support or onboarding.

However, it could also be that your lowest tier is attracting a segment of customers who aren't as good a fit for your product as the higher-tier customers. You'll have to do some more digging to find out.

Segmenting by Marketing Channel

Seeing churn based on marketing channel is an advanced approach, but the results can be eye-opening. It allows you to see which of your marketing channels (or salespeople) are driving long-term growth and which are increasing sign-ups that churn quickly.

As Aaron Kassover, founder of AgentMethods, told me: “*Segmenting churn by acquisition channel has been really helpful for us. Learning that the LTV of a pay-per-click lead is way lower than average saved us from wasting a lot of money.*”

Unfortunately, doing this requires some setup. In AgentMethods, they track it in ChartMogul by sending attribution data as a custom attribute they filter against.

As Aaron explained, "*Besides any UTM data, we also send things like sales rep, onboarding rep, [and] NPS score. It's really interesting to see how metrics like churn and ARPA vary by salesperson.*"

You could roll your own solution to this, but taking developer time away from your product is a foolish decision in my opinion. I would recommend Aaron's approach or using a tool like SegMetrics (built to solve this problem) or Mixpanel to allow you to segment your churn by marketing channel.

Segmenting by Cohort

Another interesting way to segment churn is by time-based cohort. The easiest way to do this is by setting up a retention grid, which sorts customers based on their tenure with your company and their paying relationship. Some SaaS metrics providers offer this out of the box.

MRR Retention

Month		Account Age (Months)					
		0	1	2	3	4	5
October	$3.995	99%	98%	99%	96%	92%	91%
November	$3.487	98%	96%	75%	63%	63%	
December	$3.743	101%	96%	92%	90%		
January	$2.166	108%	98%	97%			
February	$2.475	103%	106%				
March	$2.068	102%					

You'll often see that churn is significantly higher in the first one to two months of a customer's lifetime because people use that time as an extended paid trial—they are basically sticking around because they haven't yet set up their account, but they intend to.

This is helpful because if you have massive churn in the first 60 days, you can hypothesize that this is due to people not setting up their accounts or not seeing value in the product. But churn after that will often have a different cause.

To figure out why people are churning in the first few months, ask yourself:

- Is it taking too long for customers to find value?
- Are they finding that the product doesn't actually meet their needs?

You can nudge both of these in your favor.

It's Taking Customers Too Long to Find Value. If customers aren't seeing value in your product, it could be a matter of education. The standard approaches are to send onboarding emails to orient them to your product (see Val Geisler's Dinner Party Strategy) and to hire a customer success manager to walk new accounts through the onboarding process (assuming their price point makes this worthwhile).

When you have a high price point, hiring someone to help with onboarding can go a long way toward helping your customer find value.

Essentially, you're trying to help new customers find your minimum path to awesome (MPA). Basically, the moment when everything clicks and your customer says, "This is amazing!"

For a social media scheduling app, maybe it's when they load the first few posts and realize they can sit back and let your product take care of the rest. For an email product, it could be the minute they get a form installed on their website and start seeing new subscribers.

It's not always easy to find the MPA. Your product might be so complicated that there are many paths to seeing value.

In that case, the burden is on you to educate your customers about how to get the most value in the shortest amount of time. One way to shortcut the process is to interview customers who are actively using the product and ask them when they first realized how your product would help them.

With Drip, we even built a custom internal dashboard to track where trial users were along the path to awesome. Had they created their first email list? Installed a form on their site? Activated that form?

I could watch individuals or groups of users go through those steps during the trial phase and see a leading indicator of how many were likely to convert into paying customers.

Customers Are Realizing Your Product Doesn't Meet Their Needs. Sometimes this is a messaging issue. Somehow customers have unrealistic expectations about what your product will do for them, and it's not until they spend some time using your product that they realize it's not a good fit.

In this case, it's helpful to look at new customers by industry and traffic source to see if there are patterns related to how they found you or what they are trying to accomplish with your tool.

Are you overselling your product's capabilities to an audience looking for more sophisticated solutions? Is your messaging attracting a type of business (e.g., size, vertical) that isn't a good fit?

One way to answer the above questions is to simply ask.

There are many ways to ask customers why they churn. At Drip, any customer that canceled their account received an automated email within ten minutes of canceling. It said, “Hello, I’m one of the founders of Drip, and I’d love to hear why you decided to cancel your account.”

We got a wide range of responses. Some people would tell us they were shutting their business down—which isn’t something we could fix. Others would say they switched to a cheaper tool because they didn’t need our more powerful product. Others switched to a competitor because they needed a feature we didn’t have.

The key to getting useful data points out of an exit survey is to keep it short and direct, create a connection (“I’m the founder, and your feedback would help me build a better product!”), and ask for a reply—even if it’s just four or five words.

This can give you a glimpse into what potential customers want, which helps guide product decisions.

More often than not, you can get a quick win with churn using tactics like an email welcome sequence and in-app onboarding tools to make sure new users see value early. However, pushing churn below 2% or 3% is a long road that unfolds slowly as you refine your marketing and sales language, learn more about your ideal customer, and add more features those customers love (we covered that in the Market chapter).

If you reduce churn enough, you might find yourself unlocking the SaaS Cheat Code: Net Negative Churn.

SaaS Cheat Code: Net Negative Churn

Your business achieves net negative churn when your expansion revenue outpaces the revenue you lose from churning customers. While +4% churn means you’re losing 4%

of your recurring revenue each month, -4% churn means you're gaining 4% of additional recurring revenue each month, without adding new customers.

If you are able to achieve net negative churn, you have an incredible business. You can add zero customers in a month but still grow. If you continue adding new customers, net negative churn becomes a massive flywheel that helps your business grow faster and faster.

It's pretty magical to see. Not to mention that upselling existing customers is a lot cheaper than acquiring new ones, so your CAC decreases and profitability increases.

It's challenging to get to net negative churn purely by growing expansion revenue, though. You also need to do the work we talked about above to get your gross churn down to reasonable levels (in the 0% to 3% range).

When I owned HitTail, an SEO keyword tool, I structured my pricing using the value metric of website visitors. Once a customer went from 10,000 to 20,000 page views (or 20,000 to 50,000), they bumped up to the next tier.

The problem was that it didn't happen as often as I expected. And because HitTail was a fairly inexpensive tool, our churn was around 8%. Our 3% expansion revenue was a nice offset, but it didn't help us achieve net negative churn.

On the other hand, I recently spoke with a company that uses an API credit system in which the more calls a customer makes to their API, the more they pay. Like HitTail, it has a 3% expansion revenue. But because its product is high-priced and their churn is low (2%), that's enough to get them to 1% net negative churn.

Mindset

How Do I Achieve Success?

Success comes down to three factors: hard work, luck, and skill.

A particular person's journey might require an equal one-third of each factor, but in most cases, they are not evenly distributed.

If we look at Mark Zuckerberg's experience launching Facebook, we assume he worked hard and had some technical skills and a lot of luck. The luck of timing and stumbling onto an idea that others had around the same time were factors we can only guess fell into place for him.

How about Jason Fried and David Heinemeier Hansson building Basecamp into a mostly bootstrapped company that throws off tens of millions of dollars per year in profit?

When I asked Jason on stage at MicroConf in 2019 why Basecamp had over 3 million accounts created in its first 15 years, he responded, "*Truthfully, luck and timing were one and two. And we* *did a good job, too. But really it was luck and timing.*”

Hearing him recount the story, it’s obvious they worked hard and had excellent design and technical chops. Those two factors alone would have allowed them to build a great business. But maybe not one that grew organically to (presumably) more than $100 million in revenue in about a decade.

Plenty of highly skilled founders work hard on their ideas only to build companies worth far less than Basecamp.

What Can You Control?

You can control two of these factors: hard work and skill. Although we can attempt to create our own luck, we do that by working hard and acquiring or improving our skills.

Skills can be improved through applying hard work to a particular field of study, for example: copywriting, visual design, filmmaking, coding, or SEO.

With each of us starting from a different place, some may need more work to learn skills others gained through their upbringing. But you can build skills to become more successful through hard work.

You can also control your mindset.

Successful founders come in all varieties. Some are highly driven and motivated and seem to move at superhuman speeds. Others are more chill and purposeful, showing up day after day to do the work. But they all share a few key traits.

Successful Founders Have a Bias toward Action. When in doubt, they do something. They don’t wait around; they don’t procrastinate. They don’t think of reasons it should take longer—they start shipping.

A mistake I see founders make is taking too long to act. They get paralyzed in analysis and have a lot of churning without making any notable progress.

Successful Founders Develop Their Gut. I've talked a lot about "founder gut" in this book, that instinct that helps you tune out the noise and find the right path. Some founders have it innately. Others develop it by going through the School of Hard Knocks and learning through trial and error what works and what doesn't.

You can also circumvent some of the more painful lessons on your way to developing your founder gut by watching, listening, and learning from others.

You might join a mastermind group and learn from your peers. You can hire a coach. You can find a mentor—whether that's someone you know personally or someone you've never met whose work you follow. (I'll talk more about finding masterminds and mentors later in this section.)

I can't underscore enough that learning from others is a great way to develop your founder gut. You'll save so much time and heartache than if you insist on making all your own mistakes.

My founder gut has made huge leaps when I'm around someone who's amazing at something that I am not.

Successful Founders Manage Their Own Psychology. For years I've been saying that more than half of being a successful founder is managing your own psychology. I came to this realization after seeing my wife, Dr. Sherry Walling, help founders find success after months or years of failure due to mental blocks and other unhelpful patterns.

In parallel, I've observed founders who seem to succeed no matter what's thrown at them, be it unethical competitors, hackers, lawsuits, or a major platform threatening to shut them down (all scenarios I've helped founders work through).

Then there are those who get in their own way at every step. They don't do it intentionally, but they always seem to have a reason they can't ship, can't market, aren't closing sales, or aren't growing their business.

It's about their internal psychology.

In the rest of this section, I'll talk about some of the frameworks that are key to developing the mindset you need for your business to achieve escape velocity.

Let's start by helping you get out of your own way.

Where Should I Focus My Time?

In the early days, your priority is doing whatever it takes to figure out one thing: How do you build something businesses want and are willing to pay for?

There's no blueprint for this. It's a few rules of thumb coupled with experimentation and gut feelings, and because of the uncertainty, it is often a hard point on your entrepreneurial journey.

The early days are a pressure cooker. You might be putting your nights and weekends into making this work. You might be eating through your savings or relying on a spouse for support. It can be rough, which is why in those early days your priority needs to be moving as quickly as possible to get to product-market fit.

You need to move fast, and you need to work on the right things.

Before you find product-market fit, you're mostly scratching and clawing and (as Paul Graham famously said) doing things that don't scale. You might be handling email support or finding customers one at a time from your network. Spending your time prioritizing these things won't help you build a million-dollar business, but in the early days, that's what will get you to the point where you have an opportunity to build that business.

As you move toward escape velocity, you figure out how to get more of these things systematized. You hire people to handle sales demos, answer customer support emails, and build the product.

Eventually, you get to a place where the hundreds of tasks that used to be clogging up your to-do list are someone else's responsibility.

Once that happens, where should you focus your time?

Efficiency vs. Effectiveness

When I was in my 20s, I had the luxury of being able to spend a lot of time working on random things. I could come home from my day job and work 20 or 30 hours a week during nights and weekends because we didn't have children.

The problem was that I was often ineffective at what I was working on. I could get a lot of things done "efficiently." But becoming "effective" took a lot of hard work and learning.

You can be highly efficient by doing 10 tasks in a day, but if you didn't need to do eight of those, you weren't very effective. Effectiveness is when you do only the two tasks that actually drive your business forward.

A friend of mine has health issues, so he only works on his company three or four hours a day—but he gets more done than most founders do in 10 because he's not doing extraneous stuff. He has a strong sense of what will drive the business forward, and he works only on those tasks.

How do you know what to work on? Some founders are good at seeing this from the start—most of us are not. You can get better at it by:

- Being around successful founders, seeing what they work on and how they think about things
- Learning through your own experimentation, trial and error
- Exploring mental frameworks like my Stair Step Method of Entrepreneurship, Sean Ellis's Three Stages of the Startup Pyramid, and my Risk vs. Certainty framework I'll outline next

Risk vs. Certainty

In SaaS, there are risks, and there are certainties.

Certainties are things that need to get done, and you know how to do them. You know you need to build a new feature, you just need to grind for eight hours and write the code. You know you need to update your website copy. You know you need to send out an email newsletter to your customers announcing new features.

Risks are the things you're not sure of. You know you need more customers, but you don't know how to do that. You need to experiment with marketing approaches to figure that out. You know something's off with your product-market fit, but you don't know how to fix it. You need to do a deep dive into customer research to figure it out.

The areas of risk are iterative. You're rarely going to get them right the first time, so you need to try things, make mistakes, adjust course, and know when to give up and when to press on.

Risky areas require founder-level thinking. Certainties only require the ability to execute them and can be handed off to task- or project-level thinkers.

As revenue grows, you should begin hiring so you can delegate your certainties. You can hire someone to write code, write a newsletter, or file your taxes (all certainties). You should focus your energy on areas of risk because risks move your business forward, and they require a level of problem-solving that is hard, if not impossible, to hire for at this stage of your business.

Should I Raise Funding?

You may wonder why I've placed this section in the chapter on mindset. The mechanics of raising funding are straightforward, but I've found the internal struggle a founder faces when deciding whether to raise funding is often more challenging.

I've never been anti-funding. But I am against the all-or-nothing narrative that you should *always* or *never* raise funding. I find extreme absolutes are usually unhelpful, especially when deciding if you should accept outside investment.

Being anti-funding is like being anti-hammers. Funding is a tool, and you should learn when to use it and when it's not a good fit.

Ten years ago, it was much more difficult for bootstrapping founders to raise money while using a capital-efficient approach to growth and not being pressured to raise additional funding every 18 months. Most investors, whether angels or venture capitalists, share the mentality that once you raise funding, you are seeking to grow fast enough that you can raise a Series A funding round and shoot for a billion-dollar outcome.

Venture capitalists weren't interested in giving a few hundred thousand dollars to a founder whose vision involved organic, natural growth. They were looking to invest in unicorns (billion-dollar companies). The way to build a unicorn company is to raise funding, grow quickly, and repeat that cycle about every 18 months, diluting founder equity and potentially forcing a great $20-million or $30-million business to grow past its logical limit. This often implodes because the founders try to force their company into this "unicorn-or-bust" model that most venture capitalists seek.

These days, though, capital that doesn't force you onto the venture-funded track is much more accessible. It's now possible to raise an angel round or join an accelerator like TinySeed that gives you a boost without tying you to the venture-funded track.

Raising $100,000 to $500,000 after you have a modicum of initial traction won't solve all your problems, but it can certainly make things less stressful and can buy you the resource no founder ever has enough of: time.

> In your personal life, money saves you hours.
> In your business, money saves you years.

If you're working nights and weekends, raising enough to fund a year or two of your salary can be a game-changer for your company. Being able to quit your day job and focus full-time makes a night-and-day difference in achieving escape velocity.

If you're already full-time, raising funding to hire someone who can take tasks off your plate can be equally powerful. It's incredible the boost you can give your business when you're no longer stuck in the code or the support treadmill.

Funding can be a powerful tool—but you need to know what you're getting into.

Have a Plan

What will you do with the money? How are you going to deploy it to grow the company?

It is tough to find an investor who will give you money if you don't have a plan, and it can be dangerous for your business. If you don't know what you're doing, money will not fix that. You're just going to make bigger mistakes, faster.

I generally don't recommend raising funding before you have some semblance of product-market fit because (A) your valuation is lower at this stage, and (B) you're likely to burn through most of that money just trying to find product-market fit.

Don't Torch Your Cap Table

Your capitalization table is a list of who owns what percentage of your company. Literally: Rodrigo owns 70%, Janine owns 20%, and Fred owns 10%.

Your cap table can get complicated if you start taking multiple rounds of investment. I've seen cap tables with 40 entries, where the founder still owns 50% of the company, a bunch of angel investors own 5% each, and early employees each own 1% to 2%.

A complicated cap table isn't a deal breaker. But you can torch your cap table if you let early investors or founders take too much of the company. We've had multiple companies we've been unable to fund because of their cap table.

One was a company where the founder only owned 30% because he'd given up 70% to an agency he was working with in the early days. Another founder gave 60% of her company to an early investor who had only invested $50,000.

When you let early investors take too much, you end up shooting your business in the foot by making it uninvestable. You also put the majority of the profits into someone else's pocket.

You can also torch your cap table by not vesting founder equity. If you start a company with two other people and split it equally, but six months later one of your cofounders gets a full-time job and leaves, they still own 33% of your company. You and your remaining cofounder are stuck working the next five or 10 years growing a company and putting money in your ex-cofounder's pocket.

This also creates a problem if you want to raise money. Normally in first-round funding, investors want to make sure the founders who are actively working on the business own 80% to 90% of it.

This can be fixed with vesting, where you get zero shares during the first year you work at the company and 25% of shares after the first year, then the rest drip out over the next three years. Those numbers can vary—you might decide to say it's three years to vest. Just make sure to talk to a lawyer when you set it up.

Funding Is a Tool

There are a lot of dynamics when it comes to raising money, and, like I said before, raising funding won't automatically solve your problems.

At the end of the day, raising money can make the downsides of your product or business model worse. If you don't have product-market fit, haven't found a good marketing approach, or are working inefficiently, raising money can exacerbate those issues.

But raising funding also has the potential to save you years. As Craig Hewett, the founder of Castos, told me, funding allows you to "live in the future" by making investments you otherwise would have had to wait for.

When Craig Hewett raised money for Castos, he spent it on hiring senior sales and development team members rather than the juniors many startups are forced to hire because of a lack of cash. This allowed Castos to make progress fast.

Ruben Gamez, the founder of SignWell, used funding to invest in compliance (SOC2 Type 2 and HIPAA). They would have done so eventually, but they wouldn't have been able to afford it until later. This investment allowed them to start closing major deals sooner and grow faster.

Strategic hiring can be another way to spend funds. Jordan Gal, the founder of Rally, hired a chief of staff almost from day one. He told me, “*Money allows you to hire in such a way that you, as the founder, can focus on whatever your superpower is, with far fewer distractions than when bootstrapped.*”

Derrick Reimer of SavvyCal burst into a crowded scheduling space by investing funds into SEO and marketing earlier than he would have been able to if he was purely bootstrapped. This potentially shaved a year or more off his marketing efforts.

Those are just a few of the ways funding can help when applied strategically.

Drawbacks

There are three main drawbacks to outside funding.

The first is the time investment. A typical angel round can feel like a part-time job, taking 10 hours a week for three to six months. Getting accepted into an accelerator might only cost you the time to apply, handle interviews, and work with legal counsel to review and sign the docs. But there is a definite time investment involved in raising, and during those hours you won't be working on tasks that could drive your business forward.

The second is the added complexity. Anytime you add someone to your cap table, you have one more entity involved. Even if investors have limited or no rights in decision-making, you have an external source to whom you need to report financials, keep updated, and potentially obtain signatures from if you make certain changes to your corporate structure.

Finally, there's the fact that you are selling part of your company to someone else. The idea, of course, is that the funding should allow you to increase the value of your company far more than the value of the equity you sell. But that ultimately falls back on your ability to execute and grow the business with the funding provided.

Am I Turning Speed Bumps into Roadblocks?

During an episode of my podcast, as my former cohost and I dug into some struggles he was running into getting his product to escape velocity, I said this phrase: **"***It sounds like you're taking speed bumps and turning them into roadblocks.***"**

As a founder, you can choose to look at an obstacle as something that keeps you from moving forward (a roadblock) or as something that slows you down for a minute as you continue along your path (a speed bump).

As I look back on my experience building startups, I can see how my mindset around this has evolved over time. In the early days, everything seemed like an insurmountable roadblock, a company-ending event.

These days I force myself to view things as speed bumps. As I've made this seemingly simple mental shift, I've become more effective and less stressed.

Roadblocks

Problems are inevitable when you are building a company. When they arise, if you find yourself using phrases like the following, your founder lens is viewing them as roadblocks:

- This is too big to fix.
- If this doesn't work, the business is finished.
- I'm out of ideas. I can't make this work.

Obviously there are situations where a problem is an actual roadblock. A roadblock is a business-ending situation where a platform is shutting down its API or you're being sued out of existence. Although these things happen, they are much less common than speed bumps.

Speed Bumps

The tricky part about speed bumps is they masquerade as roadblocks. They do this by taking advantage of your lizard brain. The stress and uncertainty we experience as we build companies leave us vulnerable to believing things that simply aren't true.

We often do this because our stress level is high and our minds don't quite know how to handle it. When we're in a constant fight-or-flight mode, most things feel like an existential threat. They feel business-ending.

I recall trying to land a major customer for Drip, and my inner voice was saying: If you don't land this customer, the business is going to fail.

That statement was factually inaccurate. It would have been a disappointment if we didn't land the customer, but every month several significant potential customers showed up in our inbox. We had options.

This was absolutely a speed bump, not a roadblock. But I let it become a roadblock in my mind, taking a toll on my happiness and productivity in the process.

In the early days of TinySeed, I spoke with a founder who decided not to accept our investment terms. My inner monologue was telling me: Our approach is too new, and people aren't ready. It's not going to work.

Yet here we are years later with batches of successful startups I couldn't be more proud to work with. In retrospect I turned a minor rejection (a speed bump) into a mental roadblock that stuck with me for days.

Unnecessary Stress

Roadblocks put stress on your mind, on your body, and on your relationships. It's no way to live, especially for folks who are building startups to improve our lives.

We don't have to grow at all costs or answer to a board. We build startups with the handicap of not being able to raise buckets of funding. Not receiving funding means hard-earned freedom, and you shouldn't squander it by letting your mind manufacture roadblocks where there are none.

At some point a couple years ago, I realized that all the stress and worry over my then 14 years building companies was not a good thing. When people ask what my biggest regret has been as a founder, it's that I stressed too much about things that were going to work out. I made roadblocks out of speed bumps.

Optionality

My process for solving problems has moved from stressing about everything that could go wrong to mapping out three or four possible options if things do go wrong. Often these options are not optimal, but none of them would be business-ending.

Typically they involve spending more money, investing more time, turning down a lucrative deal, or navigating a sticky situation. These are all things I'd prefer not to do, but none would ruin the company.

Making the mental shift from "everything will end" to "we'll switch to plan B, C, or D" has been one of the biggest leaps in my own psychology. I've realized this over the past few years, but I heard it in full force in the podcast episode I mentioned at the start of this section.

I would not be overstating to say this has been a life-changing realization. My regret is that I didn't discover it years ago.

Where Can I Find Community?

About a decade ago, I started championing masterminds on *Startups for the Rest of Us.* They were popular in the Internet marketing space and, given the impact they'd had on my life, I felt they would be particularly valuable to both early and growth-stage startups.

Masterminds are small groups of people that have similar experiences, challenges, and trajectories in their businesses. Simply put, they embody the phrase: Two (or three) minds are better than one.

Because many bootstrapped founders run solo operations, the idea of having a crew to provide feedback—based on an understanding of our industry that few have—was too good to pass up.

There are three clear benefits of being a part of a mastermind:

Growth. By surrounding yourself with folks who can provide you with informed advice, qualified referrals, and critical constructive feedback on your failures in a safe space, you are setting yourself up with the resources and guidance you need to focus on growing your business.

Hopefully there's at least one other person on each call who has more experience in a specific area than you do. There are some things in my business that I'm pretty damn confident I'm good at. But I know I have blind spots in other areas. I can bring those things to my mastermind because they don't have the same blind spots.

Accountability. As many solo founders know, keeping yourself accountable with no outside forces can be challenging. During most mastermind meetings, there is a point in time when each member is in the hot seat, discussing past goals and their progress, setting new goals and tracking them, and reporting back to the other members with updates along the way. By asking your group to keep you accountable to your business, you're also committing to holding them accountable.

A good mastermind also forces you to look at your weaknesses. You can bring your weakest attributes in front of this small group of trusted individuals who know your story, your revenue, your growth rate, and all your foibles—personally and professionally.

Your mastermind will tell you things that if your spouse said them, you'd ignore them. (Ask my wife about that.)

Support. Humans are social beings. Napoleon Hill says that the convergence of two individual minds creates a third, invisible force that combines the strength of both of its components.

When you share your vulnerabilities and successes with others, you magnify your own experience and make it the experience of those around you. The power that comes from those shared experiences can be just as compelling and empowering as your individual success.

Three of my favorite entrepreneurial communities are Indie Hackers, the Dynamite Circle, and of course, MicroConf.

How Can I Find a Mastermind?

As the idea of masterminds has become more common in the entrepreneurship space, they're becoming easier to find. You can start one of your own—I typically recommend you join a community and look for members within that community with whom you gel.

Or you can take advantage of a matching service like MasterMind Jam or the one we started at MicroConf. We started matching because we had so many people asking us how to find a mastermind, and at the time of this writing, we're close to matching

1,000 founders across 50 countries with more than $200 million in collective ARR.

We also put together a free guide to masterminds, including specifics about format, how to form and run a group, and everything else you need to get started. You can read it at microconf.com/mastermind-101

Choosing the Right Mentor (Or Two)

While a mastermind is a group of your peers—people who are on the same path and facing similar challenges—a mentor is someone who's been down the path before. A mentor gives you advice from a place of success: they've successfully bootstrapped a startup or two, or they've been in business for years.

Note: I use the word "mentor" here unconventionally. In this context, I mean online mentors with whom you chat regularly and those you follow from a distance (through a podcast, blog, book, or YouTube) and may never connect with directly.

A mistake many new founders make is either listening to too many people or no one at all.

I made the second mistake for years. I thought I had such a unique take on business that I didn't need to listen to others who had come before me and likely had insight into how I could be successful.

Other founders make the opposite mistake, following 10 or 15 different people who each have different approaches to starting a company. They try to combine all of their information into a single approach, which often leads to information overload and a mess of conflicting tactics.

My advice: find a mentor or two. But no more. This keeps your information consumption at a reasonable level while allowing input from outside sources.

How to Choose a Mentor

Perhaps a mentor is creating content that resonates with your goals, or they've built a company you admire. Here are a few questions to consider when choosing a mentor:

- Have they accomplished what I want to accomplish? Does it seem like they got lucky, or could they do it again?
- Does this person have a reasonable personal life? Do they treat people well? Do they have a happy family? If someone has a catastrophic personal life, odds are decent that if you're not willing to do the same, you'll be unlikely to achieve success in the same way they did.
- Do you want to be like them, and are you willing to do what they've done to become successful?

Who you choose as a mentor is a deeply personal choice, but it's usually pretty obvious who resonates with you both as a founder and as a human.

How Can I Avoid Burnout?

In 2015, as I was riding the rocketship of Drip, I started to burn out.

As the founder, I was bearing the burden of every task that slipped through the rest of the team. I managed legal, HR, payroll, operations, business development, and marketing; I co-led product and engineering; and I ordered snacks for the office. These were all things the company needed, and I was fully capable of doing them. A few of them brought me joy.

Every month I asked myself how I could hire someone to do all of the above. But being cash-strapped in a hypercompetitive space meant that shipping more features felt more important than finding someone to handle day-to-day business tasks.

Some months Drip grew enough to warrant hiring one or two additional full-time employees. And every time, I opted to hire in customer success, sales, or engineering. I wanted to close more deals and ship more features, my happiness be damned.

Except it began to take a toll in a serious way. My motivation slipped over the course of many months. After overcoming the prolonged financial stress of 2014, when Drip wasn't growing quickly enough, I began to experience burnout in mid-2015. Many days I showed up to work and stared at my Trello board, unmotivated to do any tasks. Nothing on the board looked interesting.

Jason Cohen did a talk at SaaStr in 2018 that put visuals to what I experienced.

Everyone deserves fulfillment

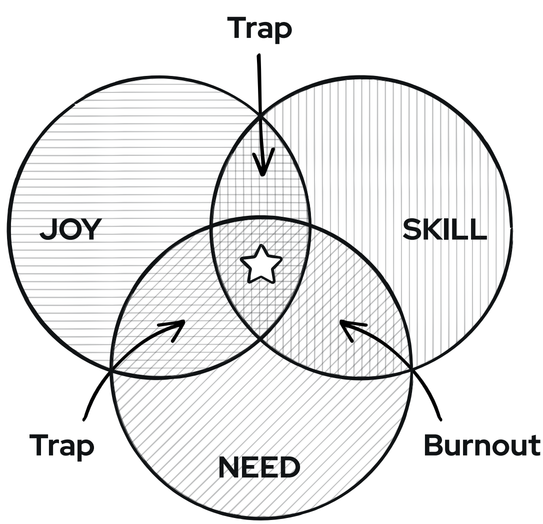


Image adapted from Jason Cohen's talk at SaaStr, 2018.

In one circle are the things you're happy doing. In the second are the things you're good at. And in the third are the things your company needs.

The sweet spot for your work should be where all three intersect. If you're focusing solely on things you're good at that bring you joy, you can get stuck galloping down paths that are detrimental to the needs of your company. If you're doing things the company needs that bring you joy (but you're not good at), then you're dragging your company down.

But if you're stuck doing things the company needs that you're good at (but don't like), that leads to burnout.

That's exactly what I was doing. I hired an executive assistant who lightened that load for a bit. She helped streamline a few things and made appointments, but what I really needed was someone to whom I could delegate at another level.

At the time, I felt like we couldn't afford someone who wasn't contributing to the bottom line of the company. In retrospect, this was one of the biggest mistakes I made while building the company.

I should have hired someone who could come into the office and handle operations. Things like legal, payroll, HR, and facilities. Most of these were outsourced to external providers, and it was just a matter of interfacing with them.

As I look back at my descent into burnout, one thing that could have saved me was having enough funding to hire someone to do the work that didn't bring me joy. Or prioritizing spending money on hiring and delegating tasks that didn't move the business forward but were contributing to my lack of satisfaction at work.

I hope you're not at a place where the next section is helpful to you. I hope that you're smarter than I was and are putting measures into place to keep yourself from burning out like I did.

As Jason said in his talk: "*The right question is what should you be doing differently now [...] in order to build a company that's more* *healthy and prosperous, and also avoid this balloon payment of emotional toil at the end.*"

Getting through Burnout

I've heard burnout manifests itself differently in different people, but for me it felt like a mix of depression and frustration. Not the kind you can clear up with a weekend away, but a long-term, deep sense of tiredness, lack of motivation, and feeling just a little pissed off at all times.

This led to my aforementioned stints of staring at Trello for hours and a lack of presence at home. I would be home with the family, but most of the time my mind was elsewhere.

I tell this as a cautionary tale: if you find yourself listless, unmotivated, or constantly frustrated as you endure the stress of building your company, it's unlikely to fix itself. The best remedy for burnout is significantly changing your habits and patterns related to work, including stepping away for weeks, which feels like the last thing you can do when everything is going crazy (whether it's good crazy or bad crazy).

If you find yourself in this situation, you need to address it, or it will get worse. If unaddressed, burnout can lead to terrible outcomes, including long-term damage to your brain.

If burnout is a situation you find yourself in, consider reaching out to a professional who works with founders on the mental game of entrepreneurship. Dr. Sherry Walling is one (she happens to be my wife, and she knows her stuff), but there are many other executive coaches and therapists who work with high-performing individuals to manage stress, burnout, and everything else that comes with our line of work.

I've only started to touch on burnout here, but for an entire chapter about it, check out my third book: *The Entrepreneur's Guide* *to Keeping Your Sh*t Together: How to Run Your Business Without Letting it Run You.*

What Are Founder Retreats?

After I left Drip, I took six months off to decide what to do next. I wasn't sure I was going to start another company.

I wanted to do something fun. I love tabletop games, so I considered getting into the tabletop game space—maybe manufacturing them, publishing them, or building a tabletop gaming site.

I was actually in talks with a major tabletop review site and was starting to look at a hard pivot. What would it look like to sell the podcast? To sell MicroConf? Maybe it was time for me to get out of the startup game and sail off into the sunset to game until I got bored of that and moved on as well.

I was on the verge of making a life-changing decision, and I needed some perspective. I decided to go on a retreat.

Taking a Founder Retreat

The two biggest things that have helped me in my journey as a founder are masterminds and founder retreats. Without those, I sincerely don't think I would be as successful as I have been.

My wife Sherry has a PhD in psychology. She started going on annual retreats after we had kids, where she got away for 48 or 72 hours without podcasts, movies, or books—just herself, a notebook, and silent reflection.

When she first started taking retreats, it didn't sound like my thing. I'm always listening to a podcast or an audiobook. I'm constantly working on the next project. But after seeing her come back from these retreats energized and focused, I decided to give it a try.

I booked myself a hotel on the coast and drove out for the weekend with no radio, no project, no kids, and no distractions.

Over the course of that two-and-a-half-hour drive, things began to settle. I started feeling everything I hadn't had time to feel for the past year. In the silence, I had sudden realizations because I was finally giving them quiet time to emerge.

During that retreat, it became obvious that my whole life had been about entrepreneurship. Ever since I was a kid, I have wanted to start a business. I've always been enamored with being an entrepreneur and the excitement of startups.

I realized that I was coming to this decision of what to do next because of the idea of wanting to get away from the thing that had caused me to feel bad—as though startups were at fault rather than the decisions I made.

At that time, my podcast had more than 400 episodes, which had been recorded over eight years. That wasn't an accident. It existed because I loved doing it. I showed up every week even though it didn't generate any revenue.

During my retreat, I realized that being involved in the startup space is my life's work. The podcast, my books and essays, MicroConf—all were part of my legacy. Instead of selling it off and striking out in a new direction, I decided to double down.

Within a couple months, I launched TinySeed. Then I leaned into the next stage for MicroConf, where we transitioned from a community built around in-person events to an online and in-person community, plus mastermind matching, virtual events, funding, and mentorship.

I also began working on this book.

As a founder, it's important to know yourself. Even if you started out with firm self-knowledge, the fast pace and pressure of bootstrapping a business—not to mention the pressures of the rest of your life—can make it difficult to see your path. A founder retreat is a way to reacquaint yourself with yourself every so often.

After my first founder retreat nearly a decade ago, I started going on a retreat every six months. Now I do one a year, and it's one of the most important things I do for myself, my business, and my family.

If you're considering a retreat, several years ago Sherry wrote an ebook called *The Zen Founder Guide to Founder Retreats* that explains exactly what questions to ask yourself, the four steps to ensuring you have a successful retreat, the list of tools she recommends bringing along, and how to translate your insights into action for the next year.

Afterword

You made it! I appreciate you sticking with me through parts that were fun and those that were less fun (metrics, anyone?). My hope is that this book brought you many times more value than what you spent buying it.

If you liked it, a rating on Amazon or a mention on Twitter would help me tremendously.

If we're not connected on Twitter, I'm @robwalling. Feel free to ask me questions about the book or anything startup related.

If you want to hear more about building, launching, and growing startups, I host a weekly podcast at startupsfortherestofus.com and a weekly YouTube series at microconf.com/youtube.

If you have a question or to purchase the print, ebook, or audio version of this book, visit saasplaybook.com.

Thanks for reading!

Acknowledgments

This is my fourth book, and trust me when I say that writing them does not get easier. This book is only possible because of the many people who generously contributed their time and energy.

Thanks to my parents, who long ago taught me the value of hard work.

Thanks to Ruben Gamez for convincing me I needed to write it and for being the place where I steal a lot of my good ideas.

Thanks to my writing coach and developmental editor, Jessie Kwak, for asking the right questions and putting up with my many rewrites.

Thanks to Charlie Gilkey for the pep talk when I stalled halfway through the book.

Thanks to Sherry and the boys for their support during the years it took me to learn these lessons.

Thanks to Ruben Gamez, Craig Hewitt, Aaron Kassover, and Derrick Reimer for offering critical feedback on early drafts.

Appendix A: Resources

Although it is far from exhaustive, this appendix is intended to provide you with additional resources for learning and community, including several that were mentioned throughout the book.

I maintain a page on this book's companion website with up-to-date information on topics that change frequently. To view the page, visit saasplaybook.com.

Communities

MicroConf is a community of tens of thousands of ambitious SaaS founders, ranging from those in the idea stage to those who have many millions in revenue. It offers both online and in-person events, a free online community (hosted in Slack), mastermind matching, and more. Learn more at microconf.com

Indie Hackers is a community of developers launching side projects and building them into full-time incomes and beyond. Indie Hackers has a long-running podcast in addition to its vibrant (and free) online community. Learn more at www.indiehackers.com

The Dynamite Circle is a community focused on digital nomads and those looking to build incredible lifestyles through their businesses.

They cater to every type of entrepreneur, from software to freelancers to agencies, e-commerce entrepreneurs, and content creators. They offer a paid online community, in-person events, mastermind matching, and more. Learn more at dynamitecircle.com

Rhodium is a community of digital entrepreneurs and investors. They offer an online community, facilitated masterminds, and in-person events for entrepreneurs running content, e-commerce, SaaS, and productized service businesses. Learn more at rhodiumweekend.com

Podcasts and YouTube Channels

Startups for the Rest of Us is a 30-minute podcast I've shipped weekly since 2010. It covers many topics contained in this book, including all aspects of building, launching, and growing SaaS companies. Learn more at startupsfortherestofus.com

Bootstrapped Web, with Brian Casel and Jordan Gal, is a long-running conversation between two SaaS founders who talk week-to-week about building their companies. Learn more at bootstrappedweb.com

Lenny's Podcast, with Lenny Rachitsky, is an interview show that focuses on building and growing great products. Learn more at www.lennyspodcast.com

In Demand, with Asia Orangio, breaks down marketing myths and lays the foundation for both bootstrapped and funded SaaS founders to grow their businesses with strategies and tactics that work. Learn more at in-demand.castos.com

MicroConf's YouTube channel is where we publish weekly videos, usually 10 to 15 minutes long, focused on tactical SaaS-related topics, from ideas to validation to launching and growing. Learn more at microconf.com/youtube

Dan Martell's YouTube channel is where SaaS coach Dan Martell talks about all things SaaS, from branding to sales to growing your team and more. Learn more at youtube.com/user/danvmartell

Masterminds

If you are interested in being matched with like-minded founders, check out these SaaS-focused mastermind matching services:

- MasterMindJam. Learn more at mastermindjam.com
- MicroConf Masterminds. Learn more at microconfmasterminds.com

You can also check out MicroConf's free guide to starting a Mastermind at microconf.com/mastermind-101

Books

Marketing

- *Traction* by Gabriel Weinberg
- *Hacking Growth* by Sean Ellis and Morgan Brown
- *The 1-Page Marketing Plan* by Allan Dib

Sales

- *Product Demos That Sell: How to Deliver Winning SaaS Demos* by Steli Efti
- *Founding Sales* by Pete Kazanjy
- *Demand-Side Sales* by Bob Moesta and Greg Engle
- *The Ultimate Sales Machine* by Chet Holmes, with Amanda Holmes

Positioning and Branding

- *Obviously Awesome!* by April Dunford
- *Building a StoryBrand: Clarify Your Message So Customers Will Listen* by Donald Miller

Mindset

- *The Zen Founder Guide to Founder Retreats* by Sherry Walling

- *The Entrepreneur's Guide to Keeping Your Sh*t Together* by Sherry Walling with Rob Walling

M&A

- *The Art of Selling Your Business: Winning Strategies & Secret Hacks for Exiting on Top* by John Warrillow
- *Before The Exit: Thought Experiments For Entrepreneurs: A Short Guide For Founders Planning to Sell Their Business* by Dan Andrews

Customer Research

- *The Jobs To Be Done Playbook: Align Your Markets, Organization, and Strategy Around Customer Needs* by Jim Kalbach
- *Deploy Empathy: A Practical Guide to Interviewing Customers* by Michele Hansen
- *The Mom Test: How to Talk to Customers & Learn If Your Business Is a Good Idea When Everyone Is Lying to You* by Rob Fitzpatrick

Other Works by the Author

Start Small, Stay Small

A Developer's Guide to Launching a Startup

Start Small, Stay Small focuses on practical, step-by-step instructions used by hundreds of developers on the road to launching their startups. Whether you have a product idea or are still looking, this book takes you through the process of finding an idea, testing it, converting visitors to buyers, and attaining profitability as quickly as possible.

"Every software engineer can pull an idea out from here... I wish I had this when I was starting my business."
– PATRICK MCKENZIE, KALZUMEUS

The Entrepreneur's Guide to Keeping Your Sh*t Together

How to Run Your Business Without Letting it Run You

by Sherry Walling with Rob Walling

Running a business is hard. It can ruin your health, your relationships, and your life. *The Entrepreneur's Guide to Keeping Your Sh*t Together* is an invaluable guide to staying sane and ensuring both you and your business thrive for years. Learn new ways to deal with the responsibility and fear of being an entrepreneur; cope with depression, anxiety, burnout, ADHD, and other common psychological burdens; and overcome procrastination to get more things done—faster.

"A personal, generous, and incredibly useful guide to staying sane and changing the world at the same time. Read it before you think you need it."
– SETH GODIN