CHAPTER 2

WHY THE STARTUP WORLD HATES ON SERVICES (AND WHY YOU SHOULDN'T)

(Consulting) is dancing with the devil as you pursue your dreams and try to pay the bills. I believe that it is near impossible to build a successful software product while maintaining a services business.

—Giff Constable, 2010

The Moz story should have ended with bankruptcy/failure/my dad shutting down the operation/me getting beaten up by Rocco the debt collector. How is it that thirteen years later, we're a 155-employee, $45-million-dollar-a-year software business?

Unbelievably enough, a side project that turned into a consulting business saved the day. The experts in the startup world will tell you that services and consulting are a waste of time. Fortunately, I hadn't yet heard that advice.

When I started working at my mom's company, our business model was exclusively service-based. Many small businesses start this way. We did projects as diverse as business card design, website usability consulting, e-commerce implementation, print media ad design (yup, like the ones in magazines and newspapers—my high school yearbook layout skills came in real handy), and, of course, search engine optimization.

The services model has some unique advantages and some frustrating drawbacks. On the plus side, costs are (supposed to be) low. You need to spend money only when you have client projects covering your outlays. Your work can be highly customized, which makes it easier to sell to disparate customers with little in common (versus the one-size-fits-all model of a product-focused company). The downside is that it's very hard to scale. The one hundredth client to whom you provide services needs just as much time, energy, and work as the fifth client (granted, you may gain some efficiencies, but it's nothing like the scale a product-based business can achieve).

We were never particularly strategic about changing our business model or thinking of ourselves as something bigger than a niche consulting firm. When, in 2002, Geraldine (yes, we dated a long, long time before getting married) asked me what I hoped Moz might become, I laid out a dream for a consulting firm with fifteen to twenty employees, a nice client list, a healthy profit margin of 20–30 percent, and some beautiful offices. I hoped that in five to ten years, we could become that company.

A couple of years later, I'd split my time between SEO work for our clients and hours chatting away on the SEO forums with like-minded practitioners around the world, trying to learn more about the practice. But I wasn't satisfied with the limited options for starting and replying to threads on other people's sites. I wanted my own platform and the freedom to publish my own way. So I spent a few nights

9780735213326_Lost_TX.indd 30 7/19/18 11:16 AM

coding up a blogging system in Dreamweaver (this was in the days before plug-and-play blog software like WordPress was popular) and, in October 2004, launched SEOmoz.org.

People ask about the name a lot. (The attempted pronunciations are usually delightful. For those of you wondering—*it's Ess-ee-oh-mawz.*) The name "Moz" was a reflection of my admiration for other "moz"-named projects on the web—the well-known-at-the-time open directory project DMOZ, the free restaurant and recipe website Chefmoz, and the open-source music discovery site Musicmoz.org. All of these names drew inspiration from the nonprofit Mozilla Foundation, itself a portmanteau for "Mosaic killer," Mosaic being the very first World Wide Web browser and Mozilla aiming to displace it with a free, open-source browser of its own. My hope was to build a similarly open-source, free, authoritative resource for the world of SEO and search engines. I'd registered it as a .org, redirecting the domain "seomoz.com" to "seomoz.org" to help communicate that a noncommercial focus was intended. A savvy adviser teased me about the .org domain extension.

"Are you in the business of being noble?" he asked. I laughed. I didn't know I was in the business of being anything.

The blog was a first step, built in my off-hours as a passion project with the goal of learning and sharing more about the weird world of ranking websites in the search engines. At the time, information about search engine operations and SEO practices was immensely hard to come by. So I enlisted help of the only kind I could afford: family.

My grandfather Seymour and I spent days poring over patent applications by Google and Yahoo!. We read technical papers from conferences on information retrieval (the science behind search engines). He taught me how PageRank, Google's famous, link-based algorithm, worked—the iterations required to get graph convergence, why a dampening factor existed, how new links on a page could siphon PageRank from the existing links, etc. His mathematics and engineering background was invaluable, and we turned a number of our technical-paper review sessions into blog posts and content pieces for the SEOmoz website.

He and my grandmother also get co-credit for helping potty train me and teaching me to ride a bike. Point is, I owe him a lot. (Thanks, Amma and Papa.)

As is true for many startups, the idea or business we eventually pursued came from a wholly unexpected place. That blog became a bootloader for the business, exposing the brand to a vast array of serendipitous experiences that led us to our eventual path. Today, it might be called "content marketing," but when I was writing, it was simply passion for sharing, a youthful craving for attention, and a hatred of the secrets Google kept that drove those nightly posts.

In the summer of 2005, I got an email from a reporter with *Newsweek* magazine, Brad Stone (who later went on to the *New York Times* and then *Bloomberg Businessweek*), who wanted to interview us for a potential story on the world of SEO.

I was thrilled and hopeful at the possibility of press coverage. Although we were gaining in popularity with search industry insiders, we were still hamstrung by debt. We had plenty of work from a handful of clients but needed to keep our expenses extremely low. Gillian had stopped taking a salary entirely. My coworker, Matt, and I were taking home a paltry $1,600/month each. The $1,000 I spent earlier that year on a Toronto speaking gig was a big drain.

That autumn, Brad flew to Seattle for a visit to our cramped office above the movie theater. I remember being deeply embarrassed about our dingy space (and the smell of popcorn that wafted up), but it didn't seem to matter to Brad. He was writing a story about the practice of SEO, something that almost no mainstream media outlet had previously covered, and wanted to feature SEOmoz in the piece. I don't think he cared one iota about the size or aesthetics of our office, with its flickering fluorescent lights, or that weird stain on the carpet where we were pretty sure a raccoon had given birth.

In December 2005, Brad emailed to let us know that the following week, *Newsweek* would be publishing a multipage article, including a large photo of me and my mom. I was thrilled. Then I panicked.

What if tens or hundreds of thousands of people visit and want to learn about SEO? The blog is too focused on insiders, and it won't be accessible to all these newcomers. They'll go search for SEO information somewhere else. I'd squandered an opportunity that didn't even exist yet!

That week I poured dozens of hours into a new project I called "The Beginner's Guide to SEO." It remains one of the most productive freak-outs I've ever had. The result was a giant document, almost a novella in length, detailing the many aspects of search engine optimization—how to do keyword research and discover what your customers are entering into the search engines, how to create content people and engines will both appreciate, how to identify problems or issues that could hinder search engines from effectively crawling and indexing your site, how to earn links and attention, the whole SEO shebang. Matt thought I was a madman. I barely made it into the office. I spent most days at home relentlessly trying to finish the guide. I even let some client work slip.

When my tome was at last completed, Matt used his design skills to polish it up and we posted the nearly forty-page guide on SEOmoz's website. We put a message on our home page welcoming visitors we hoped would see us in *Newsweek*, directing them to the Beginner's Guide if they wanted to learn more about SEO. Then, we waited.

The article came out online first, on December 11, 2005. Then it was in the print edition a week later, on December 18. Home delivery was somewhere in between. And the traffic bump we saw to SEOmoz across a two-week period surrounding it accounted for, based on our analytics, fewer than five thousand new visitors. Considering we normally received one thousand daily visitors, the article had been merely a small bump, a far cry from the hyper-growth accelerant we'd hoped for.

But that *Newsweek* piece helped us in an indirect way. The week prior, on December 6, we'd posted the Beginner's Guide to SEO in anticipation of the *Newsweek* article. Sometime on December 7, the popular technology news website Slashdot put the piece on its home page, sending more than thirty-five thousand visitors to SEOmoz in the first twenty-four hours. Dozens of other blogs and websites followed suit, sending huge amounts of traffic and attention our way. Taken together, the coverage of the Beginner's Guide dwarfed the traffic and press value from the *Newsweek* piece. My panic about how to teach *Newsweek* readers about SEO ended up as a far more effective catapult for SEOmoz's blog and brand than the audience for whom I'd written it.

That piece of content, built to help serve a handful of magazine readers, transformed us from a niche, industry-insider blog to one of the most recognized brands in the SEO space and brought us the clients we so desperately needed to survive. A few speaking engagements later, we were working with companies like eBay, Yelp, OpenTable, Zillow, and more.

As our revenue grew and the stress of just making it through the next week and dodging the debt collectors faded, I found myself frustrated by a cycle that many businesses get stuck in. We were getting . . . comfortable.

The bills got paid. The clients were usually happy. They referred us to their friends. Our salaries went up a few hundred dollars. These aren't bad things! But something was missing. I couldn't shake the feeling that our website was too popular for the consulting business to make sense. We had thousands of visitors every day, but could barely take on more than six or seven active clients in a month. Consulting is limited entirely by time and people. It took me two to three nearly full days to put together a basic site audit with recommendations. We hired a couple of recent college grads who, after a few months of training, could do the same work (with some oversight and review from me) in a week. Then there was communication overhead and keeping up with industry changes and writing on the blog and closing new deals and the surprisingly time-consuming process of turning away work that wasn't a good match. We were immensely busy, but adding to our capacity only drained us more (as every hire required closing more clients and training time and reviews). Many businesses get stuck in this cycle—not wanting to scale because it creates more work and less profit margin in the short run, thus limiting their upside in the long run.

Turning away business created this nagging sense in the back of my skull that we weren't executing on our opportunity, that SEOmoz could be more.

Luckily for us and mostly by accident, Moz transitioned from services to product. In late 2006, Matt and I hatched an idea to open up access to some of the proprietary tools we'd built to help with SEO tasks for our clients (things like tracking rankings in the search engines and checking for problems that could harm visibility). We were mostly seeking more traffic for the website but realized that having the tools available completely for free could overload our server bandwidth and cost way too much. Thus, we put the tools behind a $39/month PayPal subscription. We figured that this would help lessen the traffic overload risk and still get us more clients.

We had no idea it would, almost overnight, lead to a transformation of Moz's business model.

When the subscription to our tools opened in February 2007, it didn't feel like a momentous, overnight transformation. A few new subscribers joined every day, but by and large, business carried on as usual. Our primary focus remained centered on consulting. It was only a few months later, while analyzing our revenue, that we saw the potential of the subscription business. The run rate of our new product was growing like a weed. By the end of that year, we did about $400,000 in SEO consulting revenue (our fourth year in that business), and $450,000 in software revenue (even though we'd only had ten and a half months of it!). It was a wake-up call to the power of having a product that made money while we slept.

How to Escape the Services Hamster Wheel

Why is it that so few consulting businesses successfully launch a product? What made Moz's case unique? And why, given the success of our consulting efforts, did we decide to move to a software subscription model?

There are *two* traits fundamental to an effective product-focused business. The *first* is reach (i.e., the ability to influence a large audience). The *second* is scalability (i.e., an aptitude for growing revenue far more quickly than costs).

Traditionally, consulting businesses have little of either. Consultants don't need wide brand awareness or a large audience. They require only a small group of highly targeted individuals and organizations to be aware of their existence and services. Word of mouth is often enough to fill the pipeline for consulting businesses. In a product-based business, though, a much broader audience is typically required, and there's vastly greater need for brand awareness and market penetration. Word of mouth alone can power an exclusive list of enterprise-only product companies, but even then, the competitive landscape dictates a degree of coverage and scaled-up marketing that's almost never found in the consulting world.

Consulting businesses are rarely scalable, unless their firms are extremely well staffed (think Deloitte or McKinsey), and when they are, it's thanks to processes and people. In a product-based business, scalability comes from the adeptness of the product itself to serve a wide audience with a single offering (or a range of products that all benefit from the same design and development process). In a consultancy, hours and projects are what customers pay for rather than the goods or access (physical or virtual) that power a product-based firm.

As we looked at the differences between our nascent software subscription business and our years-old consulting business in 2007, it became immediately clear that our bias would be to the software side of things. That's not because it's right for everyone but because it was right for *us*. The things I loved to do most—helping people, writing, speaking, building community—were the marketing channels that enabled us to attract customers and serve them well. The background in consulting gave us empathy for our customers, because we'd done the same tasks they had and knew the products we (and thus, often, they) needed to better do our work. Unlike most consulting businesses, we could leverage a large community and a highly trafficked website we'd created on the side. We were committed to sharing knowledge openly with all of our competitors, and we happened to have a programmer on staff who could build software to scale our processes.

We'd unintentionally invested in many of the pillars needed to create a successful product business. Our strengths in content, community, and reach meant that even though our first product wasn't great, we had branded ourselves as trustworthy operators in a field where that had historically been rare.

But it wasn't just our strengths and passions that led us to software; the financial side had a big impact, too. The revenue we earned from subscriptions had superior gross margins, far less time investment per dollar earned, and required very little hiring or contracting compared to our consulting work. We discovered through trial and error what financial markets have known for years: the dollars earned from a recurring revenue model are vastly more valuable than dollars earned from services, thanks to scalability and margin.

Wait, So Some Money Is Worth More Than Other Money?

Financial models that value a company's revenue care a lot about gross margin (i.e., the percentage of a business's or product's income that can be serviced without additional cost). If you build software and provide it to customers, your costs are, basically, maintenance of that software, hosting of the servers, any data you have to buy to keep the software operational, and . . . not a lot else (maybe customer service and support). Conceivably, you could shut down all new development efforts and let go of most of your staff, and the subscription dollars would keep coming in. Hence, gross margins on software product businesses are often higher than 75–80 percent, while the margins on a consultancy are more likely in the 25–40 percent range.

These distinctions affect not just the business's revenue-generation process but how they're valued by potential buyers and investors, too. Every dollar you make from services will net you (on average) one to two times the amount in an acquisition or valuation scenario. This formula for a product-driven business, like our software subscription, is often in the three-to-eight-times range.

Say we have two entrepreneurs who've built similarly sized, strong businesses over the last few years and are both ready to sell and retire. Niki's firm is a software-subscription business, and Silvio's is a consulting firm. Both have fifty employees, $10 million in revenue over the past twelve months, and a growth rate of 30 percent in each of the last four years. My examples here are vastly oversimplified, but the raw, average outcomes are instructive.

Niki can reasonably expect that her company will yield, on the very low end, $30 million, and on the high end, $80 million in a sale. It could be even higher if her technology is especially in demand or if the skill sets of her engineers, the unique data she's collected, or the market she's tackling is massively interesting to multiple powerful companies.

Silvio can probably anticipate a sale of $10 million to $25 million for his similarly sized business. The modifiers will generally be his company's EBITDA (earnings before interest, taxes, depreciation, and amortization), his net margins (profitability), and some elements of relative demand, bidders, and industry.

Little wonder that investors, then, put so few dollars into consulting businesses and so many into software companies. And little wonder, too, why at Moz, once we started to understand these numbers, we quickly doubled down on software and put less energy into expanding consulting.

I'd Like Some Services, Please

Should every consultant or services-based business try to shift to a product/subscription model? Definitely not. Despite the example in the previous section, the delta in outcome for many product-based businesses isn't nearly as dramatic as the technology media makes it out to be. It's true that product companies usually get more attention and coverage—Salesforce and MailChimp have numerous write-ups in national papers and fawning descriptions of their founders' brilliance. That seldom happens to consultancies. But services-based businesses still have many great strengths that deserve consideration:

- Little to no startup capital is required; you can build these companies from home, with nothing but your own time, energy, and effort.
- You can entirely control the scale, expenses, and profitability of your business with far greater precision by taking on more or fewer clients, raising or lowering prices, and doing the work yourself versus hiring or outsourcing.
- If and when you choose, you can take time away from the business with far less harm to its long-term potential. No new clients for a month and no deliverables committed means a month with family or on vacation.
- You will almost never need to give up ownership or shares in the company, which has tax benefits (using an S corporation or LLC, which most consultancies do, means neither profits nor salaries get taxed twice) and leaves you in full control. This isn't to say that

a software/products business cannot do this, just that it's much less likely, especially if you want/need to raise capital.

- In general, hiring costs are much lower in consultancies (because most types of consultants tend to have lower salaries, on average, than software/product engineers, designers, and marketers), people retention is less challenging, and the benefits/perks are not expected to compete with those of Google, Facebook, and others of their ilk.
- Most surprisingly: services firms are often a superior financial deal for a founder.

Don't believe me on the last point? Let's go back to our comparison of Niki and Silvio:

As Niki started her software business, she raised two rounds of financing, $500,000 from angel investors in exchange for 30 percent of the company's equity, and later, a Series A round from a venture investor of $8 million in exchange for an additional 40 percent of equity. These modest investments and the 15 percent of shares in the company she put in the employee option pool and distributed to team members meant that, upon sale, she owned 15 percent of the company she started. That's slightly higher than the median startup founder, who owns only about 11 percent of his or her company's shares at exit.

Silvio, on the other hand, didn't seek outside investors and retained 100 percent ownership of his consultancy (standard for businesses of that type). So who is ahead when it's time to sell? If we presume each got the median valuation for their sale, Niki made 15 percent of $40 million ($6 million) and Silvio made 100 percent of $15 million. Despite the massively higher valuations of software revenue, Niki made $9 million less than Silvio. Here's all that in side-by-side chart form:

	SILVIO'S CONSULTING BUSINESS	NIKI'S PRODUCT BUSINESS
Investment Raised	$0	$8,500,000
% of Business Owned at Time of Sale	100%	15%
Total Revenue at Time of Sale	$10,000,000	$10,000,000
Revenue Multiplier at Sale	1.5X	4X
Total Sale Price	$15,000,000	$40,000,000
Founder's Share of the Sale	$15,000,000	$6,000,000

That is a lot of money. In fact, my friends, it's twice what the original Adam West TV series Batmobile sold for at auction in 2013. Silvio has two Batmobiles more than Niki! Which, I think we'd all agree, is the single greatest indication of wealth.

However, if as a founder of a product company you're able to retain a larger-than-average share of your business, it's a great deal. If Niki held on to 60 percent of the company she started, again presuming the two founders got the median valuation for their sale, she'd make $24 million (60 percent of $40 million) versus Silvio's 100 percent of $15 million.

The point is, you could get an extra Batmobile if things go right. Maybe two.

But this conversation is moot if the business fails. Fifteen percent of zero is bupkes. And tech startup failure rates don't apply to the services category with nearly the same ferocity. In 2012, Scott Shane of Small Business Trends analyzed US Census Bureau data and found that services firms have some of the best survival rates among small-business types: 47.6 percent make it past year 5 of operation. Contrast that with the less than 25 percent of tech startups that do. *You are, statistically, almost twice as likely to "make it" as a services business than as a technology product firm.*

But, weirdly enough, you'll almost never read about acquisitions of services firms in the tech press. You'll almost never see a seven-, eight-, or even nine-figure merger or acquisition from the consulting world make it to even the bottom of Techmeme, or get a tweet mention by a prominent figure in the startup world. The focus is almost entirely on product companies, VC-backed companies, and those few in the United States with connections to the major media outlets of tech's insular ecosystem. In my opinion, this isn't just correlated with the perception of services firms; it's causal.

I have a lot of friends and colleagues who run consulting companies. Whenever we talk finances, they're blown away that, with a few hundred thousand or million in revenue, they are often personally much, much better off financially than my wife and I.

"How can that be?!" they'll ask when I share our fiscal situation.

"Well," I say, "I have a salary at Moz. And I have my stock. But I can't pay myself any of our profits, and until/unless the company sells or goes public, my stock is illiquid."

"Damn"—they always shake their heads—"a forty-five-million-dollar-a-year business doesn't make you rich. Weird."

I'll dive more deeply into the monetary machinations of venture-backed founders in chapter 8; stay tuned.

How to Push Your Product

I don't mean to unfairly or irrationally condemn a product focus. There are a number of good reasons to choose a product business over a services one:

- Your passion is to build product, not to do consulting.
- You love marketing to a broad, diverse audience, rather than networking and building a reputation among a more niche group.
- You will not be satisfied with yourself or your accomplishments until you try the product path.
- You accept the larger risks, higher failure rates, and more intensive capital demands in exchange for the potentially larger rewards that come with a product-centric company.
- You crave the attention, press, and long-shot odds of becoming wealthy enough to make Solomon blush.

This was the case with Moz all those years ago, and so a transition to product made sense. If you're in the same boat, here's the best advice I have on how to make this transition go smoothly.

Start with a product that's informed by your consulting. The services you provide expose you to real-life problems that consumers and organizations face. You have solutions, in the form of applied knowledge, for which people are willing to pay. Your experience can inform a product's design, content, marketing, amplification, and even the early parts of audience building. Moz's consultancy certainly did that for me—it showed the real challenges organizations of all sizes faced with SEO. It exposed me to hundreds of scenarios and problems, many of them overlapping, and gave me the knowledge of how to solve them that I could share transparently with others (which built the basis for my blog). Before I ever designed a tool, I knew exactly what issues I wanted to solve, and more important, I knew there were many others like me trying to solve those same problems. Thousands of them had already commented on my blog (and, helpfully, opted in for email messages).

Build a scalable marketing practice that attracts your prod-

uct's audience. Because so much of consulting business is driven by small-scale marketing and word of mouth, it's rare to find a consultancy seeking to build a sizable audience. But this is precisely what gave Moz the ability to scale its software product after launch, and if you can leverage marketing channels for your consultancy in the short term, with an eye to your product business in the long term, you'll be vastly ahead of the game. The key is to find channels of attraction with enough overlap to accommodate both the audience you need now—services clients—and the ones you need to reach in the future—product buyers.

Use the services revenue to fund the product's creation and testing. Don't let obsession with your new idea overtake your focus on the consulting business's success. It can be tempting to devote the majority of your time and the business's resources to product development, especially when you're excited about what you're creating (and you better be damned excited, because that passion will be needed to get through the ugly parts of product scaling). But if you cut short your consulting revenue, or fail to fill your pipeline with new clients, or let your work quality suffer, your adventure could be over before it begins.

When I talk to founders who've attempted this shift, three big things are almost always holding back a successful transition.

1. Comfort with the existing model, and the reliance on services income to survive.
2. The undistracted time needed to build a great product.
3. The difficulty of finding enough of the right customers for that product. It's rare that your services clients are the perfect match for your product (because fundamentally, they need services, not a product; that's why they're your customers!). This is why you

can't simply transition services clients to product customers, despite the seeming appeal and overlap.

If you internalize these challenges ahead of time, you can form testable theories of how to overcome them. If you recognize the odds, the sacrifices, and the upsides of both models, you can wisely choose between them. And hopefully, you can learn from Moz's experiences and make use of our tactics to ease a potential pivot.

Consulting Isn't the Enemy; Biased Thinking Is

So which model is right for you? Let's directly compare the two:

	SERVICES BUSINESS	PRODUCT BUSINESS
Startup Costs	Low	High
5-Year Average Survival Rates	Mediocre (47% according to US Census data)	Miserable (<10% according to NVCA data)
Marketing Demands	Low	High
Limits to Growth	People, Marketing, Retention	Capital, Engineering, Customer Acquisition
Competitive Barriers to Entry	Low	Medium to High
Scalability	Low	High
Staffing Requirements	High Ratio of Staff to Customers	Low Ratio of Staff to Customers
Gross Profit Margin	Low	High
Net Profit Margin	Medium	Low (often due to reinvestment for growth)

Clearly, the answer to the product versus services debate is "it depends." Smart founders need to use their best judgment about their strengths, their goals, and their market to determine whether one model is a clear winner, and/or whether it pays to experiment with some of each. Unless you're expressly seeking to raise money from venture investors, nothing's stopping you from offering a subscription product to your services clients, or from adding consulting services to your product business. Don't let Silicon Valley culture's traditional thinking on this bias you away from the right decision for your company and your customers.

CHAPTER 3

GREAT FOUNDERS DON'T DO WHAT THEY LOVE; THEY ENABLE A VISION

Your work is going to fill a large part of your life, and the only way to be truly satisfied is to do what you believe is great work. And the only way to do great work is to love what you do.

—Steve Jobs, 2011

Can I geek out for a minute?

I love SEO.

I love how small changes to a web page can make a marked difference in how it appears in search engines and how that drives hundreds or thousands of people to visit my site. I love the combination of technical skills and creativity required to overcome the competition for a tough keyword. And I love the mystery of how search engines rank pages and the process of uncovering each little piece of that puzzle.

When I find a new tactic or uncover a subtlety in Google's ranking process, my eyes light up. I do my best, deepest work, going for hours on the pure delight of discovery and intrigue, desperate to prove my hypothesis about what's happening in the millions of mathematical calculations that drive visibility on the web. If I get enough evidence, and can repeat the experiment consistently with ranking success, I'm elated. I'll run around my apartment, fists pumping in the air like a scrawny, nerdy, Jewish Rocky Balboa (avert your eyes, neighbors; not all our windows have blinds). I'll spend hours more writing up and visually documenting my work for a blog post or a presentation. When I finally hit "publish" or step onstage to share my findings, it's the pinnacle of my professional endeavors. Sharing that knowledge and removing that shroud of opacity from how these systems that govern what billions of people experience every day through search is what I love to do.

So, obviously, as CEO of an SEO company, I should be happy as a clam, right? I'm living the (admittedly, nerdtacular) dream, aren't I?

CEO Is a Real (Shitty) Job

Early in my career as an SEO consultant, a majority of my days were spent actually doing the work I loved. But after the company started on a serious growth trajectory and I assumed the role of CEO, I probably spent less than 20 percent of my time doing those things, and it often dwindled to less than 5 percent for months at a time.

We grew fast. And it was my first time doing anything (and everything) a CEO has to do. The learning curve was steep and uncomfortable. I didn't just have to learn; I had to immediately apply that knowledge and iterate it until I got it right. The stakes were high. Employees were relying on me to assign the right projects that would help make our software better or help us reach a broader audience. Customers were relying on me to build and support SEO tools that were better than their manual alternatives (or our competitors' tools). Investors relied on me to hire staff, to execute on projects, to report how things were going, to maintain financial discipline, and, most important, to grow fast. And, of course, our community of hundreds of thousands of marketers relied on me to investigate, educate, and publish nightly about the search and web-marketing fields.

No big deal, right? (Cue hyperventilation.)

I remember a single week in October 2009 when I was recruiting a new CTO, grasping at the desperate end of a failing fundraising process with a number of Silicon Valley VCs, preparing presentations for two upcoming conferences, negotiating new salaries and stock compensation with senior members of our engineering and operations teams, designing the wireframes for a major new product initiative (Open Site Explorer, which would go on to be one of our most popular products), promoting a book I'd coauthored for O'Reilly Media (*The Art of SEO*), and, unbelievably, meeting with some senior tech staff from the United Nations to talk about how search visibility might be helpful to them.

(Also, figuring out what I was going to be for Halloween. I ended up going as a red-shirted ensign. You know, the guys who are first to die tragically on *Star Trek*? Let's not read into that.)

I did virtually no hands-on SEO.

The myth of "founding a startup so you can do what you love" is at least as enshrined in the tech world's popular culture as the myth of getting rich. It deserves to be unpacked and examined. There's a grain of truth that lies within, albeit a grain buried under layers of maddening falsehood.

Passion Does Not a Manager Make

If you didn't have a preexisting notion of what startup entrepreneurship looks like, it might seem logical to imagine a series of events something like this: the entrepreneur graduates college; works for a year or two in a field that provides broad exposure to many different types of businesses (perhaps consulting); gets an MBA; does a thorough analysis of potential opportunities in a variety of fields; identifies the one with the least competition and highest demand potential; creates product, marketing, and scaling plans; fundraises; and then starts a company.

Hell, it sounds reasonable to me.

But we know from data, and from experience, that most entrepreneurs, and an overwhelming majority of the most successful entrepreneurs, don't start out with anything close to this level of deliberate evaluation. Instead, we dive headfirst into the thing we're passionate about, without even considering the alternatives, the market risks, the competitive landscape, the long-term demand curves and macroeconomic forces that might indicate it's a terrible time to start a new science-of-pasta-making website that relies entirely on cheap ads for revenue (but maybe someday we'll also make our own pasta and sell it on the site). Because, damn it, we love pasta! And we want to share our passion with a world of people who love it, too, but don't know what to buy or how to cook it. People who have yet to learn the handful of little secrets that transform *cacio e pepe* (a Roman pasta with cheese and black pepper) from a five-ingredient bowl of mush to the most satisfying, addictive, and unbelievably delicious dish you can make in twelve minutes flat.

Entrepreneurs start out doing what they love. Not because it makes sense, or because it's a great market, but because they cannot imagine themselves doing (or eating) anything else.

On the one hand, that passion and commitment is an asset. It can help early-stage companies push through the ugly barriers that make it so difficult to find a business model. On the other, once there's a small, working operation that's found revenue, the leadership needs to refocus on all the tasks a growing organization demands. Testing ingredients, taking photos, finding obscure, forgotten recipes, and sharing them with the world may have gotten you where you are, but it won't take you from food blogger to media empire.

When your startup is growing, the tasks and competencies change every six months. From 2007 to 2014, the most important things I did were never the same for longer than that. There were parts of each new job I enjoyed, but I clung for far too long to elements of work I should have delegated, reasoning, as almost every entrepreneur I've talked to does, that it was a core competency for the business. That I could do it better than anyone else. And that I could handle that work plus my other responsibilities.

I held on to near-complete control of our blog, using the logic that my posts were what had built the company's reputation, even though diversity of content and more of my freed-up time could have seriously bolstered our product and engineering. I maintained ownership and reviews of our consulting work (which we kept up until 2010). I insisted on being our primary product designer, and on having final say over every nook and cranny for years, which held back growth from other team members and slowed down product improvements dramatically, too.

Unless what you love is managing people, handling crises, delegating, holding people responsible, recruiting, setting, then constantly

M 9780735213326_Lost_TX.indd 53 7/19/18 11:16 AM

amplifying and repeating the company's mission, vision, strategy, and values, being a startup CEO may not provide you with the work you love to do.

Instead, a startup provides the ability to create a vision you love and to see it through to fulfillment. You get to say "today, the world works this way, but once the company I'll build exists, and once it reaches the scale to fulfill its ongoing mission, the world will change to this." If you can reset your passion from "I want to do *this* work" to "I want to see something I create change the world in *this* way," your expectations will align with reality, and the cognitive dissonance and frustration of being torn away from the work you love can fade.

Note that "change the world" doesn't have to mean "change the *whole* world." Your mission could be to make available and accessible the delicious, neglected pasta recipes of Emiglia-Romana. It might be to give the people of Seattle a better, more fulfilling experience with yoga. Or it might be to eradicate student loan debt and class stratification worldwide by building an affordable educational platform with the rigor and brand respect of Harvard at the cost of a Netflix subscription (side note: please, someone, do this). But if your mission is to "do the work I love doing without letting the business get in the way," I strongly recommend against pursuing growth that demands more people on your team. Hiring people adds organizational complexity, the mortal enemy of heads-down, passion-focused, deep work.

How to Be a CEO in 6 ~~Easy~~ Infuriatingly Challenging Steps

If you're ready to accept this limitation in the work you'll get to do, and to put shepherding your vision ahead of doing what you love, you have a new set of challenges ahead. Great startups aren't built by people who never gain competence beyond their particular field—they demand proficiency in financial strategy, task planning, human resources, conflict resolution, office management, fundraising, customer support, payment collections, business intelligence, and dozens of other functions founders rarely consider in the early stages. You'll get to learn each of these through a process the startup world affectionately calls "muddling through." I prefer the more accurate description: "wading into a painful slog of failure, learning, and repetition."

In essence, you'll:

1. Realize, frustratingly late, a particular pain point that's holding your team back from effectively or efficiently accomplishing a goal (e.g., you can't run the A/B tests on your website because you have no framework for measuring results, nor any expertise in how to build effective tests).
2. Attempt a variety of techniques to overcome that pain point (hiring people with previous competency in the practice, researching and learning it yourself, delegating the learning to other team members, acquiring technology or data, implementing a rigid new process, etc.).
3. Determine that most of the things you've tried have failed, and cycle through many more until you . . .
4. Experience the elation of breaking through, at least partially, and finding some success.
5. Uncover new frustrating side effects and unintended consequences of your solution(s).
6. Half the time, settle for partial implementation and a set of compromises that work well enough to become the new norm; the other half, realize that the solutions are worse than the problem and determine a way to simply avoid or ignore the pain point,

possibly by refocusing your strategy so you simply don't have to deal with it.

I'll share an example from Moz.

From 2004 to 2012, project planning followed a very informal path at the company. I'd sit down with one or a handful of our team members, talk about what they could be working on, what they wanted to work on, what we thought would be best for the business, and through those weekly or monthly conversations, establish a road map. Sometimes we'd change it over an email thread or through a hallway chat. And sometimes we'd do a decent job of broadcasting the plans to the rest of the company. But things were small enough (at least when we had fewer than thirty people across our four functional teams) that this informal process mostly worked.

But by 2010, the forty(ish) folks inside Moz got frustrated that knowledge about what each person and team was working on was mostly held in a janky combination of (a) Rand's memory banks, (b) random email threads, and (c) a wide variety of project-tracking systems, some of which were used by only one person. The frustration amplified when projects would cross teams and drag people off their individual tasks to collaborate at inconvenient or unexpected times. To combat this, we tried a quarterly cadence of all-hands meetings, during which I or the department head would run through all the projects on the calendar for the next two to three months. This alleviated some pain but created plenty of new ones.

So we tried a series of weekly emails: each team would send out their in-progress projects and next set of expected work each Friday. By the time we had broad adoption of this practice, there was a massive amount of all-staff emails going out each Friday, most of which were ignored by most people in the company. Complaints about this glut of no-one-reads-them emails led to a consolidated process where each week, one technical project manager (TPM) would get all the teams' reports and put them into a single, very long, Friday email. If you were diligent enough and had the patience, you could scroll through that list and see what everyone was doing and what they had planned next. It basically only worked well for one person—me. I love email, I'm good at it, and very often, I'd be the only person replying with questions and clarifications on those threads.

As you might imagine, that practice neither scaled nor worked well for anyone else. When Sarah Bird become Moz's CEO in 2014 (more about that to come), she implemented a new system of quarterly planning meetings with facilitated discussions between teams over two to three days in a large room filled with poster boards on which each team's projects were listed out for discussion and markup. For those few days, representatives from each team, along with managers and executives, would huddle together and talk through every project and plan, determine work sequencing, and argue over priorities, and many times Sarah herself would have the final say about which piece mattered most.

First to go were the outside facilitators, which many, myself included, felt were inauthentic and created more complexity and process than necessary. Then the meetings shrunk to one and a half days as additional elements of discussion and negotiation were pared back. And, finally, after realizing that the process was simply too demanding and didn't provide enough value to justify its ongoing existence, Sarah eliminated it altogether. She reorganized the company into functional business units, with a smaller number of shared services departments, and shifted to a model where broadcasting everything everyone was working on was no longer required. Each team was given more autonomy to determine its own road map, and less reliance on other teams to get their work done. The pain subsided. The pace of progress increased. And despite (or perhaps, by virtue of) not always knowing what everyone else was doing, Moz's employees felt less frustrated, less overwhelmed, and more focused.

I'm skipping over a ton of the incremental steps, hard decisions, and tedious changes designed to address the struggles that arose throughout this process, but the narrative remains. It's probably familiar to anyone who's worked in a scaling organization, and it's critical to understand if you're going to build a company designed to grow. It took six years, more than ten iterations, and two CEOs to find a process that worked for product planning at Moz. In the end, Sarah made the tough but smart choice to avoid the problem altogether by restructuring the company.

The Best Leaders Know When to Lead—and When Not To

Many founders believe they can delegate key business functions to other people on their team—and often they should. But your first hires in any of these roles will need guidance, support, input, and occasionally you'll need to dig into the details yourself because you can't find the right person or you had to let them go, or you simply need to know more about the issues before you can determine whether the problem lies with your people, your management, your processes, or something else.

I hired multiple CTOs to run the technology side of our business, but never got to know the processes well enough to support or mentor a technology leader. Sarah did likewise, but finally buckled down and decided to dig into the role herself. We actually fought about it . . . a lot. I was angry that she wasn't recruiting a CTO after we lost

Anthony, our fourth engineering leader in six years. She countered that until she spent time understanding and leading that part of our business herself, she didn't feel confident in hiring an effective outsider. It was a point of contention for more than a year between us, but she proved to be totally right. Her insider knowledge of the teams and people, the work needed, the relationships and conflicts proved instrumental in finding a great tech leader, hiring him, and then eventually promoting him to the position of CTO.

Sarah and I both delegated a set of work we weren't cut out for (neither of us have particularly technical backgrounds), but we didn't get it functioning smoothly until she invested the time to understand the practice herself.

This is the work entrepreneurs do in growing organizations: digging into problems, untangling conflict, freeing people from the mind-sets or structures that hold them back, crafting (and refining, over and over) the pillars and policies of how the company functions. The hours you spend on the business will shift from doing what you love to enabling a vision and navigating whatever impediments arise along the way. Expect to do work you don't love in order to allow what you do to flourish. If you don't, the disappointment and frustration can kill your motivation.

Embrace that reality, though, and you'll come to see the CEO role as one of enabler and problem solver. For those who love helping others get unblocked and watching progress scale far beyond what they could achieve alone, this can be an immensely rewarding job.

CHAPTER 4
BEWARE THE PIVOT

Ideas are worthless. Execution is everything.

—Scott Adams,* 2010

There's an idea floating in the mythos of Silicon Valley's hallowed halls that the "pivot" is a fundamental right bestowed upon all startups, designed to help absolve the sins of your past incarnation and allow you, too, to go from Tote to Pinterest, Odeo to Twitter, or Glitch to Slack. If the decisions upon which you founded your business prove foolhardy, never fear, the pivot will save you! It's more important to get started than to spend months evaluating and choosing a wiser path. The path can never be known! The path must be discovered by trying and failing and . . . you guessed it, pivoting.

* Fittingly, most of Scott's own ideas about ethics, politics, and humanity are complete crap. Don't read his blog unless you're ready to watch a talented cartoonist reveal himself to be a shitburger of a person.

Bollocks.

In the annals of startup history, there are tens of thousands of companies that achieved remarkable success—hundreds of millions or billions of dollars in returns, a lasting impact on their ecosystems or industries, delighted customers and users, and a financially well-compensated founding team that now spends their days swimming in a pit full of gold coins (side note: everything I know about the lifestyles of the wealthy comes from Scrooge McDuck).

But among all those successful companies, how many great "pivots" were there? If we're using the formal definition of migrating completely from one business idea to a radically different one, all my research could uncover only a few dozen. Slack, Flickr, Twitter, Pinterest, PayPal, Groupon, and Instagram aren't just among the most famous, they're among the *only* ones (at least of those that have achieved truly lofty, founders-now-have-a-gold-coin-filled-swimming-hole success).

This should come as no surprise. Pivots don't happen on a whim. You change your business model, your product, your market, or your entire idea only if things are going very poorly indeed. Anything else would be foolhardy (if it ain't broke, don't pivot). It's nasty, ugly, hard, grueling work building these things in the first place, and if you've achieved any progress whatsoever, you're likely to stick with it, learn, and improve.

Given this reality, it might pay to be less cavalier and more analytical in your approach to choosing an industry, an idea, a product, and a target customer. It may also pay to choose a field others ignore because it's perceived as unsexy, sketchy, or uninteresting by some other vanity-centric logic.

In my case, Moz picked a field experiencing massive growth (being found in search engines during Google's dramatic rise) and a model that scaled wonderfully (software to help marketers and SEOs do their work more efficiently, with better results), and we spent our energy iteratively improving. We got better at marketing, at building software, at collecting data, at designing product interfaces, at retaining subscribers, and at all the other things core to our business. We favored improving execution over changing our model, market, or fundamentals. From 2007 to 2013, that brought us 100 percent year-over-year growth, leadership in our space, a variety of accolades, and the adoration of our customers and investors.

Execution Is More Malleable Than Market, Model, and Idea

I started writing about SEO in 2003. Back then, very few people were doing so, and even fewer were doing it in a transparent way that helped others learn the practice. There was a pervasive belief in the field that sharing too much about how search engines worked or what tactics earned rankings could put you out of business. Many SEO consultants perceived their "secret sauce" knowledge to be more important to their clients than the work itself.

That secrecy, combined with the unpopularity of the industry, gave my blog a unique advantage—it stood out among the available offerings for its open, transparent approach to the subject matter. The writing certainly wasn't anything special. The advice itself wasn't vastly superior to what others had. But while our competitors were mostly unwilling to share this kind of information without a signed NDA and a consulting fee, SEOmoz.org required nothing but a browser visit. As a result, other sites linked to me. Press outlets cited my posts. Consulting clients found their way to our contact page. And thousands of industry practitioners, hungry for knowledge, signed up for daily updates.

Fast-forward to 2007, when we launched our subscription business. The tools contained there weren't especially great. They were often broken or overloaded. The data they gave back was mediocre at best. But digital marketing was a fast-growing field. SEO was heating up as a practice. And there were almost no other sources of automation for folks who couldn't write their own software, so Moz stood out.

In both of these cases, we were mediocre providers who, over time, dramatically improved our offerings. My writing in 2004 was pretty sad (you can still find those early blog posts on Moz.com/blog and see for yourself). It was at least a few years in before I was producing anything that could be called "high quality." Similarly, when we launched our software via a $39/month PayPal subscription, the tool set was lackluster, the features subpar, and the usability just a hair better than atrocious.

What did we do right?

We picked a good market. We stumbled onto a compelling communication medium. And we chose a good business model. These inadvertently wise choices covered up a tremendous number of mistakes and a steep learning curve.

I published no fewer than one thousand blog posts before my posts achieved consistent, broad readership that earned the kind of value we see with published content today. From late 2003 to early 2007, I wrote for an hour or a few, five nights a week, Sunday through Thursday, and I'd spend additional hours the next morning promoting the posts, replying to comments, and finding new topics to cover. By the time the blog appeared on "must-read" lists, I'd invested hundreds of hours researching and writing. I started out an amateur but today have a massive following and the ability to generate tens of thousands of visits, resonant messages, and powerful business impacts via my blogging.

The software side of Moz has uncanny parallels. Our first tools were barely worth the subscription prices. But over many years, we got better at software development processes, at affording and hiring great engineers, and at designing more useful applications. From a few dozen subscribers to tens of thousands, from an average customer lifetime value of a couple hundred dollars to more than $2,000, we improved in every facet of the SaaS model (Software-as-a-Service, wherein subscribers are charged on a recurring basis to access software via a website or Internet-connected app).

We love to praise execution, as if executing well on any dumb old idea would take us somewhere. Sure. Immense dedication, skill, and the hard work of great people can overcome most obstacles. But choosing wisely at the start—the field, the approach, the customer target, the economic model, and the marketing methodologies—has a massive impact on the difficulties you'll face and how forgiving the journey will be. Not everyone can afford the costs of starting over. Not everyone has the privilege of being able to test hypotheses willy-nilly. If you have a family, if you have debts, if your cost of failure is anything but zero, it makes better sense to tread carefully.

The Switching Costs Can Kill You

Here's the weird thing about this argument in favor of pivoting: execution is far more fungible than your idea, your business model, the industry you choose, or even your team.

What happens as a startup makes progress? The team improves the quality of its work. Customer service folks improve their response times. Product features and functionality catch up to customer needs. Engineers deliver better technology. User experience goes from bare bones to impressive. The marketing funnel widens. Conversion rates go up.

If you're prioritizing execution and learning from your mistakes, you're already doing this.

Now imagine how hard it is to move from targeting one market to another. Many of the hard-won lessons your marketing team or salespeople or business development folks have earned are useless. You're back to square one on how to attract customers, how to close deals, how to retain their loyalty. Yikes.

Say you switch your idea or product. You'll throw out months or years of sweat and toil validating a concept, earning customer buy-in, and attracting influencers, press, and perhaps even investors. Maybe this new product or service will be easier to build than your last one. But it's not free. And it's certainly going to set you back on every other vector—including customer traction and acquisition.

Or maybe you're changing from one business model to another. It may be a smaller shift than the two above, but it still takes a vast degree of energy, and likely means migrating your customers (if they'll come) from one system of compensation to another.

Don't believe the hype—execution isn't everything. You can be the tortoise, rather than the hare, and by picking the right race and the right route, win over far more talented teams because you're constantly improving in a less crowded space no one else has chosen.

Some Unorthodox Tips on Choosing Your Market and Your Idea

If you haven't already read Eric Ries's book *The Lean Startup*, go do that now. Then pick up *Sprint* by Jake Knapp and the Google Ventures team. The first one will help you nail the basics of choosing and validating a market, and the second shares my favorite method for nailing new products and features.

Now that you're analyzing competitors' UVPs* like a boss, and rattling off sarcastic product/market misfit jokes with the best of 'em, I've got a few additional suggestions:

1. If you can keep your ego and your aspirations below the need to pursue venture capital in order to aim for a billion-dollar unicorn, you can ignore a lot of the advice about choosing a giant, fast-growing market ripe for "disruption." Instead, it's totally cool (and may make your life vastly easier) to chase after smaller markets where you have unique knowledge and passion, and where ongoing, smaller amounts of innovation can separate you from the pack. There are thousands, maybe millions, of opportunities like these with dramatically fewer venture dollars and Harvard MBAs (or Stanford CS dropouts) chasing them. Because let's face it: most of us aren't going to be innovators in humankind's quest to explore outer space. But we might be able to develop a better way to clean out a garbage disposal than, say, sticking your hand down it.
2. Great ideas and products are often born from mediocre ones. The keys are time (enough to iterate and evolve into something remarkable), humility (enough to see what's wrong and admit a

* Unique Value Propositions

failure so you can move forward), and survival (a profitable services business can be a godsend here).

3. Your business will be even more likely to succeed if the market you target is served by incumbent solutions that are some combination of (a) hated by their customers, (b) unwilling or unable to evolve with their customers' needs, (c) protected by competitive advantages you can unravel (or that a shift in market dynamics or regulation is unraveling for you), or (d) in their early stages and not yet dominant (i.e., a non-mature market). If you find a market with two or more of these (think vacation home rentals before Airbnb or crowdfunding before Kickstarter), your odds rise exponentially.
4. Keyword research (wherein you uncover what words and phrases people are searching Google for and in what quantities) will almost always uncover untapped opportunity. Move beyond the solution keywords, and look for searches that indicate problems—the quantity of monthly searches for "cityname+taxi" helped Uber figure out which cities to launch in, just as the monthly searches for "best restaurant in cityname" helped Yelp pick their expansion markets. Two good tools for this—Google's AdWords program (you don't need to buy ads; just sign up for free) and Moz's Keyword Explorer (a shameless plug, but it really is the best tool out there).

All that said, if you have to compromise on several elements (and you almost certainly will), I'd urge you to sacrifice market size, lack of (or weak) competition, and sales and marketing tactics, in that order, before you endeavor to tackle a field where your abilities don't create a competitive advantage and a unique value proposition. It's that important.

CHAPTER 5

STARTUPS CARRY THEIR FOUNDERS' BAGGAGE

Writing code? That's the easy part. Getting your application in the hands of users, and creating applications that people actually want to use—now that's the hard stuff.

—Jeff Atwood, March 2010

It's probably no big surprise to hear that a company inherits its founder's attributes—whether they be good or bad. Install a misogynist as CEO, and you'll find that the company has misogynistic practices (cough—UBER—cough). Back a founder with self-worth issues and they'll often overcompensate through political power plays and a lack of sharing credit. Study enough startups and you'll see this pattern over and over. Amazon inherited Jeff Bezos's passion for logistics just as surely as his thriftiness with employee pay and benefits and proclivity for causing burnout. Craigslist reflects founder Craig Newmark's near-luddite innovation sensibility alongside his desire for inclusivity. Slack has Stewart Butterfield's focus on visual design, user experience, and delightful Easter eggs baked into numerous features.

One of my favorite examples of this is Jessica Mah's startup: inDinero.

In 2008, Jessica enrolled in UC Berkeley's Computer Science program. She had a passion for coding, but a couple of years in, knew she was in the bottom cohort of her class. It's not that she lacked the skills or intelligence to be a great engineer, but she (to use her words) "lacked the patience." Jealously, she watched classmates attract six-figure offers (with sizable stock bonuses) from Google and Facebook. Resolving to leverage the skills she did have—a penchant for writing, for attracting press, and for inspiring people with her explanations—Jessica started a blog (http://jessicamah.com/) where she wrote about software development, team building, and startup culture.

Paul Graham, the founder of Silicon Valley's most famous startup accelerator, Y Combinator, and a demigod in the eyes of many entreprenerds (what? I can make up dorky portmanteaus), was one of her early readers. He found her writing and invited her to apply to the program.

As Jessica put it: "It was literally the day after graduation that I went to Mountain View and started at YC. I have high standards for myself, and for people around me, but I wasn't focused on my coursework. I hate being told what to do. I started inDinero because I value freedom and flexibility."

Over the next six years, inDinero grew from two people to two hundred, raised $20 million in angel money (Jessica abhors the venture model and turned down multiple offers to raise VC), and became a media darling in the B2B startup world.

I asked her what the hardest parts of growing inDinero were, and her response was immediate: "People management. Mentoring. I hate it. I don't have the patience for it.

"When we were ten people," she continued, "I thought we wouldn't make it. I hated coaching and giving feedback and being patient while people made mistakes. As we got bigger, I got rid of all my reports except two; I do as little management as possible and it's worked great."

The other big struggle for inDinero was marketing. While Jessica's a great writer and evangelist for the business, she never found a groove with web-marketing channels like search, social, or content. But she was able to leverage her immensely compelling interpersonal skills, alongside her contrarian leanings, to build a powerful PR machine. InDinero's been featured in dozens of publications, and Jessica herself has been interviewed for hundreds of articles over the years (I do an excited dance every time I see her on the cover of some national tech or financial magazine). Each new piece is an opportunity to expose her business to exactly the kinds of customers (and influencers of customers) that inDinero's chasing.

Jessica struggled in the early days, just as many founders do, because her skills and proclivities had giant gaps with what her business needed. But despite nearly failing multiple times (including once when she and her cofounder were forced to spend their personal savings, down to the last penny, to keep the business afloat), inDinero survived and thrived. How? By Jessica's gaining awareness of her strengths, weaknesses, quirks, and motivations and then structuring her startup to work with (or around) these attributes.

The near collapses of inDinero and the actual collapses that many startups suffer are often because founders don't understand themselves and how their companies inherit these traits. If you can identify and balance (or work around) the DNA that founders pass on to their startup children, you can build on your strengths and avoid many of the journey-ending pitfalls.

The Outsized Influence of Founders

Moz has always been marketing-centric. For us, the easy part is getting millions of visitors to our website who care about the problems we're trying to solve and are seeking answers to questions about SEO. These are the very people our products serve, and we're one of the best-known, most-respected, and most-trafficked destinations for this challenging-to-reach B2B audience. Marketing has never been our problem, at least never for long stretches. The hardest nugget for us to crack has always been the product itself and the technology underlying it.

We've struggled, since inception, to create high-quality software. Perhaps when this book is published, that will have changed (or maybe giant penguins will have taken over the earth. I mean, honestly, either would be an improvement). Certainly, we've made progress over the years, but it has never been fast enough or high-quality enough to outrun far less-funded, less-experienced, less-well-known, and less-well-regarded competitors. I've literally had professional SEOs who want desperately to be our customers tell me they can't wait for Moz to build or improve upon a particular feature so they don't have to give another player in our field their money. With loyalty and brand preference so strongly in our favor, I figured it would be easy for us to hire a few more developers and build software our customers loved.

Almost every founder believes this to some degree: hire the right people to bolster your weaknesses and you can focus on your strengths.

Cut to grainy film of me, sitting alone in a dark room. My eyes are bloodshot. I shake my head sadly, as I stare into a glass of whiskey.

"If only it were that easy," I whisper.

Er, sorry. It's easy to get melodramatic about this stuff.

The thing is, it is absolutely true that a balanced team, in which one person's weaknesses are covered by another's strengths, can be tremendously successful. What's not well understood, and usually only gets uncovered after years of operations, is how founders' attributes instill themselves with near-permanence in an organization, while the attributes of the supporting team fluctuate over time. Partially, this is because founders tend to be around much longer, exerting more influence over more time. The average tenure of an employee in the startup world is only about two years. But even if you retain that supporting team, there's an undeniable, indelible imprint sourced from the founders' biases, the structure of the business they created, their recruiting, their delegation, their assignment of resources, their passions, and their blind spots.

Time and again, I see this pattern play across companies of all sizes, industries, and makeups. Founders (and CEOs) determine not just the personality and culture, but the fundamental strengths and weaknesses that govern an organization's trajectory for years or even decades. Take, for example, Virgin brands, which bear a striking resemblance in tone and style to Richard Branson himself, but also carry his particular strengths (willingness to take risks, strong brand marketing, an affinity for youth culture, customer experience as the cornerstone of competitive advantage) and weaknesses (prone to short-term thinking, unconcerned with underlying technological or product innovation versus cosmetic upgrades, inconsistent financial underpinnings).

It should come as no surprise, then, that Moz's founders—me and my mom, Gillian—had no formal programming or software development experience or education. We fell into the software world through our marketing backgrounds and exposure to the field, leaned on our writing and business development skills, and managed to hire a few great folks early on who helped us build some revolutionary software. One story illustrates this perfectly.

"Never Leave Me" Is a Weird Thing to Say to an Employee

Geraldine had a classmate in high school named Ben Hendrickson. Ben is an unusual guy. He's six four but weighs about 160 pounds. His professional demeanor is equal parts programming savant and absentminded professor. Think Ichabod Crane meets Anthony Michael Hall.

In 2007, just after we'd raised our first round of funding from Ignition, Geraldine connected me with Ben. At a Greek restaurant near our offices, over avgolemono soup and gritty coffee, I explained my outlandish dream to him of building a web index to mimic Google's. Every other person I'd talked with about this plan dismissed it as impossible or, at least, impossible without the hundreds of millions of dollars Google had at their disposal. But Ben inclined his head, stared into space for an uncomfortable few minutes, and then replied in his booming baritone, "I think I can do that."

He started solo. First, by crawling, indexing, and storing all of the Wikipedia website (which, by itself, has tens of millions of pages). Then, in early 2008, once the prototype was proven to work, he and I recruited a second engineer, Nick Gerner, through a friend who worked at Google with Nick's wife. That friend—Vanessa Fox, creator of Google's Webmaster Tools program—was someone we consulted on the web index concept. Ben, Nick, and I sat down with her over Indian food, described our plans, and listened as she explained why it couldn't work and would be folly to try. Undaunted, the three of us sequestered ourselves in the dark back room of the Moz offices and, for another six months, worked on the project we'd code-named "Carhole" (a reference to an old episode of *The Simpsons*. Moe the bartender suggests Homer is an elitist for using the term "garage," because "carhole" was just as good. The name perfectly encompassed what we were trying to do: create a less fancy version of Google's index that was just as good). I advised on the product's structure, design, and outputs, while Ben and Nick did the hard work of building the crawl, indexing, and data-serving infrastructure, as well as most of the front-end application.

We were burning cash from the investment at a healthy clip on salaries, hosting, and operations. Our revenue was growing, too, but not nearly as fast as expenses. The Carhole project (renamed "Linkscape" at launch, and later "Mozscape"), was, we hoped, our ace in the hole. We knew our customers desperately wanted the competitive information about who was linking to whom on the web, and that Google had, a couple of years prior, removed that information from their search engine (historically, you could use the command link:websitename.com in Google Search to see the links Google knew about pointing to a particular website or page). We bet heavily that giving this information back to marketers and website owners would be a big driver of paid subscription growth.

October 7, 2008. I woke up in New York, excited to launch our new project at a major search marketing conference in the city (SMX East). I went downstairs to the hotel restaurant, where a crowd was gathered around the television at the bar. Many of them were clearly nervous or entirely distraught. Lehman Brothers had collapsed. Banks across the United States and Europe were waking up to their exposure to credit-default swaps in the housing market. Stock markets were crashing. My email inbox had messages from a handful of reporters who'd agreed to cover our product launch that day, canceling their interviews at the conference, explaining that they had bigger issues to cover. I walked through midtown to the conference center dreading what might happen.

The conference was a blur, but the launch, at least inside our tiny pocket of the SEO world, was a hit. Customer signups accelerated. Our revenue grew. That December, we had our first profitable month since taking investment. Ben and Nick added a third member to their crew, Chas Williams, who'd built a prototype project in college that Moz bought from him and added to our suite.

It was the startup dream come true. Raise money to build a big, challenging piece of technology everyone said was impossible, finish on time and within budget, launch, and attract real customers who used what you built. We celebrated with the company's first holiday party. We put on funny hats and fancy clothes (okay, fine, it was more like ill-fitting sport coats with shiny shirts . . . we were young). We took photos and drank champagne. We invited our investors, and they actually turned up to celebrate with us!

For the next two years, our product got better, and our revenue grew at a fast clip. But in 2011, Nick left the company. A year later Ben got an offer he couldn't refuse from Google. Chas left not long after. We'd hired other engineers to help manage the link index project, but made only incremental gains and suffered a lot of challenging problems as we tried to grow the index's size and freshness. Over the next five years, we invested near-insane amounts of effort, dollars, and engineering time trying to improve, but fell short. We hired more than a dozen folks to work on just this one project. Some worked out and stayed, others didn't. Parts of our software improved during that time, but our link data stagnated.

Simultaneously two competitors in the market—one, a secretive operation based in the Ukraine and Singapore called "Ahrefs" (pronounced "A. H. Refs"), and the other, a British firm founded by a passionate Russian engineer whose initial goal had been to build an alternative to Google's search engine called "Majestic"—grew to market dominance. After years of leading the industry, Moz became an also-ran in the field of link data.

What the @#$% happened? We were the pioneers and now we were being outdone? We'd done the impossible but now we couldn't maintain it? Is this why Usain Bolt didn't run marathons?

I've spent years agonizing over this failure. Night after night in bed or at my computer wondering how things went so wrong, what I did to cause it, how it could have been avoided. I spent hours in meetings with engineers after another quarter or two of work produced subpar results, trying to unpack our missteps and figure out how things went sideways. But I'm not a software engineer, so I couldn't even properly assess what had gone wrong. It was like trying to diagnose an illness without a medical degree. On an alien. With three heads.

Retrospectives, analyses, comparisons, competitive intelligence, anger, sadness, frustration. And the worst one of all . . . helplessness.

Jeff Atwood said that writing code was "the easy part." Maybe for you, Jeff. I'd say it's been the hardest, most mind-numbingly difficult part of Moz. For me, marketing, reaching our target audience, and getting our product into the hands of customers has been the easy part. That's a language I speak.

Impressing them with the product we provide? *That* was the nightmare.

I was lucky when I found Ben, Nick, and Chas. Their success at building a product others told us was impossible gave me a false sense of confidence. I thought I could concentrate on my strengths and hire to fill Moz's weaknesses. That didn't work out. Through multiple CTOs and numerous engineering teams of varying sizes and compositions, we were never able to recapture that magic Ben and crew brought to the link problem.

Well, not until we got them back.

In 2016, five years after Ben had left Moz, he'd founded his own startup, Idina, alongside his prior compatriot Chas Williams. Ben and I had breakfast together after I'd made a number of introductions to help him sell his fledgling company. It turned out that Ben had exactly the opposite challenge that I did—marketing and customer adoption were nonexistent despite impressive software under the hood.

We talked about the potential offers, and about the technology he'd built with Chas at Idina. Upon hearing Ben describe the data ingestion and processing system, I wondered aloud if Moz should be making a bid for the company. Ben stared off past me, *hmm*'d for a few awkward moments, and then replied: "Well, have you solved the link scaling problem yet?"

"I only wish."

"Then, yeah, Moz should probably buy us."

"You know . . . you could have told me that before I introduced you to all these other bidders."

Fast-forward three months and, via a creatively structured (and lucrative) acquisition, the band was back together again. Ben, Chas, and the Idina infrastructure were part of Moz. Another year later and we were, once again, leaders in the field of link data, with an index to rival our competitors and a lot of happy customers.

I cannot say for certain that if I'd had an engineering background, and counted big data software design and execution among my personal strengths, that Moz would have maintained its leadership and quality in that field. And I don't know for certain that if Ben and Chas had been more experienced marketers, whether their startup would still have struggled to attract users. But I know that these are not isolated stories. They're practically universal.

It's Hard to Pick the NFL's Draft Order if You've Never Played Football

Every founder (or set of founders) has a different take on the hardest parts of building a company. And those same founders will have different takes on the easy parts. Talk to two talented software engineers who founded a company and you may find that recruiting, managing, or marketing are perceived as the most difficult issues. Talk to the marketer and people manager who founded a company down the street, and you may hear just the opposite.

The uncanny truth is that those "hardest parts" and "easiest parts" say less about the challenges and more about the strengths and weaknesses of the founders themselves. We all believe the problems and experiences we face are the most common ones, the ones every founder must struggle against. It's inherent in the psychological principle known as availability heuristic bias.

And while that phrase is daunting (seriously, do not try saying it while drunk), the concept is a pretty simple one: our exposure dictates our perception. If you've ever tried convincing someone whose politics differ from your own about the statistical reality of something they've experienced personally, you know the power of this principle. (Just think about how many times you've had a miserable experience at a restaurant only to find that the 238 Yelp reviews are all five-star.)

The conventional wisdom that you should bolster your weaknesses with great hires who can give you that strength isn't totally wrong, but every time this advice is passed on, it should contain these three caveats:

1. Lacking deep knowledge and understanding of an area means that you are less likely to have connections in that field, less likely to identify right versus wrong hires in that field, and less likely to successfully recruit and convince great talent to join. You might not even realize which knowledge you lack (unknown unknowns, amirite?).
2. A founder's weaknesses are often baked into the company's DNA and create a figurative kind of debt (nonideal practices or systems) that must be addressed before progress can be made. If you lack engineering skills, this often manifests as technical debt that must be remedied through re-architecting and rebuilding core systems before adding features or enabling scalability. If you lack people management skills, it's likely organizational debt that requires months of digging into interpersonal and intra-team conflicts, letting go of some staff, rehiring, and creating processes for engagement and teamwork that build trust.
3. When you rely on someone (or several someones) to bolster a weakness, their departure from the organization creates risk that the wound will reopen. This risk is greater in smaller and less-experienced teams where the senior leader is often the glue keeping things together with their presence, and lessens as organizations grow (so long as that leader has created consistent quality through redundancy of great people and great processes).

Hiring or finding a skilled cofounder is not the only way to bolster a weakness.

If you have great confidence (or considerable fear) that an area of weakness could be your company's downfall, you can invest in that attribute. But first, you have to know it exists.

That process is eminently achievable, but few entrepreneurs have the self-awareness or take the time to diagnose. I've fallen prey to this too many times. My advice, therefore, is to be prescriptive and deliberate in examining your own strengths and weaknesses. They will, almost always, map shockingly well to those of your company's.

Make a list of the primary functions in your organization. Now apply the following scale to those functions based on your personal aptitude for each:

Level 1: Theoretical knowledge (or less)—You have friends who've worked in the field, and in theory, understand the raw elements of the practice through lots of reading and a strong degree of interest in the subject. But you've never directly worked in the field (occasional volunteering or pro bono assistance doesn't count), never managed anyone directly who was responsible for this sort of work, and haven't done any formal training, either.

Level 2: Managerial knowledge—At this level, you've had direct reports who did this work, and completed a few successful projects. You've overseen retrospectives, heard the conversations, been involved in some tough decision making, and reviewed a lot of results. But you can't do the work yourself, at least not with any degree of certainty or confidence. You have to rely on what others tell you is achievable and the reasons others give you for why something worked or didn't, and it'll be hard to call bullshit with credibility.

Level 3: Practical, applied knowledge plus working experience—You've done this work yourself, possibly alongside a team. And though you may have graduated to managing others who did most of the heavy lifting and details, you could still pick apart and identify pitfalls, potential issues, poor assumptions, or mistakes. You may have completed classes or gone through formal training, though in general, hands-on experience is the hallmark of this level.

Level 4: Deep, working expertise plus ability to teach—You've not only done the work yourself but managed people doing it, achieved high-quality results consistently over multiple years, and taught these practices to others.

Let's use software engineering as an example because it's something so many founders and would-be founders in the startup world struggle with (just look at the thousands of questions on startup-focused forums that begin with "I want to start XYZ but am not technical . . ."). Assuming you're like me, and this skill is currently a level one or two, your options to make this a strength are straightforward:

1. Learn the process and do it yourself.
2. Start the company with cofounders who have this strength already.
3. Invest in the knowledge necessary to hire, retain, focus, and manage great talent in the field.

Nearly every piece of advice I've ever seen ignores the last one and focuses on the first two. But after Sarah Bird became CEO, I watched her go from the low end of level one to the top end of level two, and turn a fundamental weakness we'd had at the company throughout my tenure (with only occasional success stories like the early link index) into a core strength in her first eighteen months leading the company.

How did she do it? I asked Sarah to contribute so we could hear directly from the source:

> Before Moz, my only software engineering experience was a Computer Science 101 course many years before. My lack of technical depth made me feel insecure about managing engineering leaders and teams. It was pretty easy to discern when development wasn't going well, but it often wasn't clear why. Was it because I had the wrong CTO? The wrong engineers? Perhaps the technical problem we were trying to solve was harder than we imagined? Maybe we didn't have enough resources? Or the right resources? Were we underinvesting in development infrastructure and tech debt? I can spin myself in circles trying to understand the why.
>
> Here are the top things I did to become a better leader of technical people and teams:
>
> - Ask questions during skip levels (meetings where a senior manager meets with team members who report to the managers below them) with engineers that go beyond identifying problems and into concrete solutions. For example, asking, "What good development practices did your last company have that you don't see in play here?" surfaces best practices.
> - When you find an engineer on the team who can articulate different strategies she has tried, and the why behind them with conviction and clarity, give her power to try implementing some of those changes here. Be her champion, even when she ruffles some feathers making change. If some of the strategies don't work out, praise the effort.

- Read everything you can about different engineering cultures and best practices. Read dev blogs for companies like Spotify, Netflix, and Airbnb. There are a lot of great books, blogs, and conference talks about best practices. Invest time to consume as many of these as possible. For example, I follow Edmond Lau (a software engineer who worked at Google and Quora, and runs the EffectiveEngineer.com blog) and Jez Humble (who works at UC Berkeley and runs ContinuousDelivery.com) pretty closely. Go beyond the slogans into the meetings, habits, and platforms high-performing teams employ.
- Take notes about the technologies and practices that you hear about in design reviews and one on ones. Then go back to your computer and google them like crazy. Read everything you can about the technology. While reading, keep in mind that engineers are famous for indulging in so-called religious wars about why a particular technology is "so much better" than another one. Learn to spot it when an engineer moves beyond advocacy to naive devotion to a particular tech. Take everything with a heaping spoonful of salt. You need to know the lovers, the haters, and the companies that are using the tech. There's no such thing as the "perfect solution," and beware anyone who tells you otherwise. It's all trade-offs.
- Recruit technical leadership with a teaching orientation. The best way to ensure that your CTO is going to make you a better CEO is to hire a CTO who likes to teach. Make it clear that you're looking for someone to drive change and educate you and the team. Beware CTOs who try to

> "shield" you from the details. Being able to explain complex things simply is a job requirement.
>
> Ultimately, leading technical teams comes down to curiosity and courage. You must be humble, ask questions, and read a lot about engineering. If incremental changes aren't getting better results, you need the courage to change technical leadership. Keep changing and trying new things until you start seeing positive momentum, and then get out of the team's way.
>
> **Sarah Bird, CEO of Moz**

This is the power of knowing your weaknesses, personally and organizationally. You can invest in them. Double down on them. And benefit your entire company in the process.

If You've Got Strong Roots, Might as Well Grow Tall

Thankfully, founders don't just come with weaknesses (despite what our brains tell us while we're trying to fall asleep). A founder's strengths and passions will often become the organization's strengths. Smart organizations should use this to their advantage by crafting a business model, a team structure, a product, and sales/marketing channels that lean on these strengths and minimize (but don't ignore) the weaknesses.

We did this unintentionally at Moz, but with great results. We'd created a community of hundreds of thousands of marketers, a website with millions of monthly visits, and an authoritative, trustworthy voice in the industry. It would be foolish, given that strength at the top of the funnel, to have an enterprise-exclusive offering that served only a tiny swath of our audience. Instead, we used a self-service model, with a low friction signup and onboarding process, to attract thousands of free trials to our software each month.

Moz aligned the strength of community and high volumes of relevant traffic to a business that maximized the value given and received. Recall that we'd started as a consulting business, serving only a tiny fraction of our potential audience (literally a half dozen clients a month at most). Transitioning to a free-to-try software subscription gave us massive improvements in gross margin, growth potential, and connection between our strengths and our structure.

Know Thyself: Not Just a Biblical T-Shirt Slogan

There's a crucial prerequisite needed to double down on your strengths or combat your weaknesses as a founder, team, and business: self-knowledge.

Tragically, most of us have a poor understanding of our own strengths and weaknesses. For me, this lack of self-awareness led me to grow muttonchops, to teach myself how to drive a stick shift while watching exactly one YouTube video in a Belfast hotel room, and to start a software company with virtually no programming experience.

Short of nearly killing yourself on winding Irish roads or going massively into debt, how do you figure out your weaknesses? I've listed a few of the most useful ways below. If only I'd known of them before I grew out those damn sideburns.

- If you're a founder, make a list of the previous successes and failures you've had in your career, and of the elements of running a business with which you're familiar and comfortable. Chances are high that your weaknesses will be the items not on that list.

- Keep a running record of successes and failures. A shared document where people and teams record initiatives and investments that kicked butt or went off the rails won't be useful right away. But over time, you can categorize and analyze these and recognize useful patterns.
- When problems arise on a team, a project, or an area of investment, do you/could you (or your founders) personally step in to provide the remedy? When you/the founder(s) do the work in this arena, does the issue generally get fixed, or does your position get stronger? If the answers are consistently "yes," you've found a strength. If it's "no," you've got a weakness.
- List functional areas of the business (ignore how you're actually subdividing the teams/people) and record which teams or roles have high turnover versus strong retention, which have an easy time recruiting versus a tough slog getting qualified applicants. These recruiting and retention numbers have strong correlation with strengths and weaknesses.
- If you haven't yet started your business or are in the early stages, use your strategic plans to identify hits and misses. Chances are, some aspects of the plan are fully formed, with high confidence, and a strong road map of tactics. Others will have a lot of hand waving and hopeful eventualities, with the details left to be filled in by either the people you hope to hire or a model of execution, learning, and iteration. That latter group will almost certainly be what trips you up.
- Ask. Question your team. Your investors. Your friends and loved ones. Your customers. Your past managers and coworkers. So often we fail to see ourselves clearly, leaning too far in the direction

of pride or false humility (sometimes both at once on different aspects). If one person mentions an attribute of yours as a virtue or a failing, consider it. If multiple people mention the same one, you've got a more concrete answer. And remember: the more receptive you are, the more honest people will be.

These tactics are not exhaustive, and they might not uncover all the risks or opportunities you'll need. Vigilance and ongoing reflection are equally essential and equally difficult. Those same strengths a founder shows in the early years of a company may turn to weaknesses at scale, just as practice and experience over years of failure and learning can turn a weakness into a strength.