

# 8

## PIVOT (OR PERSEVERE)

Every entrepreneur eventually faces an overriding challenge in developing a successful product: deciding when to pivot and when to persevere. Everything that has been discussed so far is a prelude to a seemingly simple question: are we making sufficient progress to believe that our original strategic hypothesis is correct, or do we need to make a major change? That change is called a pivot: a structured course correction designed to test a new fundamental hypothesis about the product, strategy, and engine of growth.

Because of the scientific methodology that underlies the Lean Startup, there is often a misconception that it offers a rigid clinical formula for making pivot or persevere decisions. This is not true. There is no way to remove the human element—vision, intuition, judgment—from the practice of entrepreneurship, nor would that be desirable.

My goal in advocating a scientific approach to the creation of startups is to channel human creativity into its most productive form, and there is no bigger destroyer of creative potential than the misguided decision to persevere. Companies that cannot bring themselves to pivot to a new direction on the basis of feedback from the marketplace can get stuck in the land of the living dead, neither growing enough nor dying, consuming resources and commitment from employees and other stakeholders but not moving ahead.

There is good news about our reliance on judgment, though. We

are able to learn, we are innately creative, and we have a remarkable ability to see the signal in the noise. In fact, we are so good at this that sometimes we see signals that aren't there. The heart of the scientific method is the realization that although human judgment may be faulty, we can improve our judgment by subjecting our theories to repeated testing.

Startup productivity is not about cranking out more widgets or features. It is about aligning our efforts with a business and product that are working to create value and drive growth. In other words, successful pivots put us on a path toward growing a sustainable business.

## INNOVATION ACCOUNTING LEADS TO FASTER PIVOTS

To see this process in action, meet David Binetti, the CEO of Votizen. David has had a long career helping to bring the American political process into the twenty-first century. In the early 1990s, he helped build USA.gov, the first portal for the federal government. He's also experienced some classic startup failures. When it came time to build Votizen, David was determined to avoid betting the farm on his vision.

David wanted to tackle the problem of civic participation in the political process. His first product concept was a social network of verified voters, a place where people passionate about civic causes could get together, share ideas, and recruit their friends. David built his first minimum viable product for just over \$1,200 in about three months and launched it.

David wasn't building something that nobody wanted. In fact, from its earliest days, Votizen was able to attract early adopters who loved the core concept. Like all entrepreneurs, David had to refine his product and business model. What made David's challenge especially hard was that he had to make those pivots in the face of moderate success.

David's initial concept involved four big leaps of faith:

1. Customers would be interested enough in the social network to sign up. (Registration)
2. Votizen would be able to verify them as registered voters. (Activation)
3. Customers who were verified voters would engage with the site's activism tools over time. (Retention)
4. Engaged customers would tell their friends about the service and recruit them into civic causes. (Referral)

Three months and \$1,200 later, David's first MVP was in customers' hands. In the initial cohorts, 5 percent signed up for the service and 17 percent verified their registered voter status (see the chart below). The numbers were so low that there wasn't enough data to tell what sort of engagement or referral would occur. It was time to start iterating.

	INITIAL MVP
Registration	5%
Activation	17%
Retention	Too low
Referral	Too low

David spent the next two months and another \$5,000 split testing new product features, messaging, and improving the product's design to make it easier to use. Those tests showed dramatic improvements, going from a 5 percent registration rate to 17 percent and from a 17 percent activation rate to over 90 percent. Such is the power of split testing. This optimization gave David a critical mass of customers with which to measure the next two leaps of faith. However, as shown in the chart below, those numbers proved to be even more discouraging: David achieved a referral rate of only 4 percent and a retention rate of 5 percent.

	INITIAL MVP	AFTER OPTIMIZATION
Registration	5%	17%
Activation	17%	90%
Retention	Too low	5%
Referral	Too low	4%

David knew he had to do more development and testing. For the next three months he continued to optimize, split test, and refine his pitch. He talked to customers, held focus groups, and did countless A/B experiments. As was explained in [Chapter 7](#), in a split test, different versions of a product are offered to different customers at the same time. By observing the changes in behavior between the two groups, one can make inferences about the impact of the different variations. As shown in the chart below, the referral rate nudged up slightly to 6 percent and the retention rate went up to 8 percent. A disappointed David had spent eight months and \$20,000 to build a product that wasn't living up to the growth model he'd hoped for.

	BEFORE OPTIMIZATION	AFTER OPTIMIZATION
Registration	17%	17%
Activation	90%	90%
Retention	5%	8%
Referral	4%	6%

David faced the difficult challenge of deciding whether to pivot or persevere. This is one of the hardest decisions entrepreneurs face. The goal of creating learning milestones is not to make the decision

easy; it is to make sure that there is relevant data in the room when it comes time to decide.

Remember, at this point David has had many customer conversations. He has plenty of learning that he can use to rationalize the failure he has experienced with the current product. That's exactly what many entrepreneurs do. In Silicon Valley, we call this experience getting stuck in the land of the living dead. It happens when a company has achieved a modicum of success—just enough to stay alive—but is not living up to the expectations of its founders and investors. Such companies are a terrible drain of human energy. Out of loyalty, the employees and founders don't want to give in; they feel that success might be just around the corner.

David had two advantages that helped him avoid this fate:

1. Despite being committed to a significant vision, he had done his best to launch early and iterate. Thus, he was facing a pivot or persevere moment just eight months into the life of his company. The more money, time, and creative energy that has been sunk into an idea, the harder it is to pivot. David had done well to avoid that trap.
2. David had identified his leap-of-faith questions explicitly at the outset and, more important, had made quantitative predictions about each of them. It would not have been difficult for him to declare success retroactively from that initial venture. After all, some of his metrics, such as activation, were doing quite well. In terms of gross metrics such as total usage, the company had positive growth. It is only because David focused on actionable metrics for each of his leap-of-faith questions that he was able to accept that his company was failing. In addition, because David had not wasted energy on premature PR, he was able to make this determination without public embarrassment or distraction.

Failure is a prerequisite to learning. The problem with the notion of shipping a product and then seeing what happens is that you are

guaranteed to succeed—at seeing what happens. But then what? As soon as you have a handful of customers, you're likely to have five opinions about what to do next. Which should you listen to?

Votizen's results were okay, but they were not good enough. David felt that although his optimization was improving the metrics, they were not trending toward a model that would sustain the business overall. But like all good entrepreneurs, he did not give up prematurely. David decided to pivot and test a new hypothesis. A pivot requires that we keep one foot rooted in what we've learned so far, while making a fundamental change in strategy in order to seek even greater validated learning. In this case, David's direct contact with customers proved essential.

He had heard three recurring bits of feedback in his testing:

1. "I always wanted to get more involved; this makes it so much easier."
2. "The fact that you prove I'm a voter matters."
3. "There's no one here. What's the point of coming back?"<sup>1</sup>

David decided to undertake what I call a zoom-in pivot, refocusing the product on what previously had been considered just one feature of a larger whole. Think of the customer comments above: customers like the concept, they like the voter registration technology, but they aren't getting value out of the social networking part of the product.

David decided to change Votizen into a product called @2gov, a "social lobbying platform." Rather than get customers integrated in a civic social network, @2gov allows them to contact their elected representatives quickly and easily via existing social networks such as Twitter. The customer engages digitally, but @2gov translates that digital contact into paper form. Members of Congress receive old-fashioned printed letters and petitions as a result. In other words, @2gov translates the high-tech world of its customers into the low-tech world of politics.

@2gov had a slightly different set of leap-of-faith questions to

answer. It still depended on customers signing up, verifying their voter status, and referring their friends, but the growth model changed. Instead of relying on an engagement-driven business (“sticky” growth), @2gov was more transactional. David’s hypothesis was that passionate activists would be willing to pay money to have @2gov facilitate contacts on behalf of voters who cared about their issues.

David’s new MVP took four months and another \$30,000. He’d now spent a grand total of \$50,000 and worked for twelve months. But the results from his next round of testing were dramatic: registration rate 42 percent, activation 83 percent, retention 21 percent, and referral a whopping 54 percent. However, the number of activists willing to pay was less than 1 percent. The value of each transaction was far too low to sustain a profitable business even after David had done his best to optimize it.

Before we get to David’s next pivot, notice how convincingly he was able to demonstrate validated learning. He hoped that with this new product, he would be able to improve his leap-of-faith metrics dramatically, and he did (see the chart below).

	BEFORE PIVOT	AFTER PIVOT
Engine of growth	Sticky	Paid
Registration rate	17%	42%
Activation	90%	83%
Retention	8%	21%
Referral	6%	54%
Revenue	n/a	1%
Lifetime value (LTV)	n/a	Minimal

He did this not by working harder but by working smarter, taking

his product development resources and applying them to a new and different product. Compared with the previous four months of optimization, the new four months of pivoting had resulted in a dramatically higher return on investment, but David was still stuck in an age-old entrepreneurial trap. His metrics and product were improving, but not fast enough.

David pivoted again. This time, rather than rely on activists to pay money to drive contacts, he went to large organizations, professional fund-raisers, and big companies, which all have a professional or business interest in political campaigning. The companies seemed extremely eager to use and pay for David's service, and David quickly signed letters of intent to build the functionality they needed. In this pivot, David did what I call a customer segment pivot, keeping the functionality of the product the same but changing the audience focus. He focused on who pays: from consumers to businesses and nonprofit organizations. In other words, David went from being a business-to-consumer (B2C) company to being a business-to-business (B2B) company. In the process he changed his planned growth model, as well to one where he would be able to fund growth out of the profits generated from each B2B sale.

Three months later, David had built the functionality he had promised, based on those early letters of intent. But when he went back to companies to collect his checks, he discovered more problems. Company after company procrastinated, delayed, and ultimately passed up the opportunity. Although they had been excited enough to sign a letter of intent, closing a real sale was much more difficult. It turned out that those companies were not early adopters.

On the basis of the letters of intent, David had increased his head count, taking on additional sales staff and engineers in anticipation of having to service higher-margin business-to-business accounts. When the sales didn't materialize, the whole team had to work harder to try to find revenue elsewhere. Yet no matter how many sales calls they went on and no matter how much optimization they did to the product, the model wasn't working. Returning to his



leap-of-faith questions, David concluded that the results refuted his business-to-business hypothesis, and so he decided to pivot once again.

All this time, David was learning and gaining feedback from his potential customers, but he was in an unsustainable situation. You can't pay staff with what you've learned, and raising money at that juncture would have escalated the problem. Raising money without early traction is not a certain thing. If he had been able to raise money, he could have kept the company going but would have been pouring money into a value-destroying engine of growth. He would be in a high-pressure situation: use investor's cash to make the engine of growth work or risk having to shut down the company (or be replaced).

David decided to reduce staff and pivot again, this time attempting what I call a platform pivot. Instead of selling an application to one customer at a time, David envisioned a new growth model inspired by Google's AdWords platform. He built a self-serve sales platform where anyone could become a customer with just a credit card. Thus, no matter what cause you were passionate about, you could go to @2gov's website and @2gov would help you find new people to get involved. As always, the new people were verified registered voters, and so their opinions carried weight with elected officials.

The new product took only one additional month to build and immediately showed results: 51 percent sign-up rate, 92 percent activation rate, 28 percent retention rate, 64 percent referral rate (see the chart below). Most important, 11 percent of these customers were willing to pay 20 cents per message. Most important, this was the beginning of an actual growth model that could work. Receiving 20 cents per message might not sound like much, but the high referral rate meant that @2gov could grow its traffic without spending significant marketing money (this is the viral engine of growth).

	BEFORE PIVOT	AFTER PIVOT
--	--------------	-------------

Engine of growth	Paid	Viral
Registration rate	42%	51%
Activation	83%	92%
Retention	21%	28%
Referral	54%	64%
Revenue	1%	11%
Lifetime value (LTV)	Minimal	\$0.20 per message

Votizen's story exhibits some common patterns. One of the most important to note is the acceleration of MVPs. The first MVP took eight months, the next four months, then three, then one. Each time David was able to validate or refute his next hypothesis faster than before.

How can one explain this acceleration? It is tempting to credit it to the product development work that had been going on. Many features had been created, and with them a fair amount of infrastructure. Therefore, each time the company pivoted, it didn't have to start from scratch. But this is not the whole story. For one thing, much of the product had to be discarded between pivots. Worse, the product that remained was classified as a legacy product, one that was no longer suited to the goals of the company. As is usually the case, the effort required to reform a legacy product took extra work. Counteracting these forces were the hard-won lessons David had learned through each milestone. Votizen accelerated its MVP process because it was learning critical things about its customers, market, and strategy.

Today, two years after its inception, Votizen is doing well. They recently raised \$1.5 million from Facebook's initial investor Peter Thiel, one of the very few consumer Internet investments he has made in recent years. Votizen's system now can process voter identity in real time for forty-seven states representing 94 percent of

the U.S. population and has delivered tens of thousands of messages to Congress. The Startup Visa campaign used Votizen's tools to introduce the Startup Visa Act (S.565), which is the first legislation introduced into the Senate solely as a result of social lobbying. These activities have attracted the attention of established Washington consultants who are seeking to employ Votizen's tools in future political campaigns.

David Binetti sums up his experience building a Lean Startup:

In 2003 I started a company in roughly the same space as I'm in today. I had roughly the same domain expertise and industry credibility, fresh off the USA.gov success. But back then my company was a total failure (despite consuming significantly greater investment), while now I have a business making money and closing deals. Back then I did the traditional linear product development model, releasing an amazing product (it really was) after 12 months of development, only to find that no one would buy it. This time I produced four versions in twelve weeks and generated my first sale relatively soon after that. And it isn't just market timing—two other companies that launched in a similar space in 2003 subsequently sold for tens of millions of dollars, and others in 2010 followed a linear model straight to the dead pool.

## A STARTUP'S RUNWAY IS THE NUMBER OF PIVOTS IT CAN STILL MAKE

Seasoned entrepreneurs often speak of the runway that their startup has left: the amount of time remaining in which a startup must either achieve lift-off or fail. This usually is defined as the remaining cash in the bank divided by the monthly burn rate, or net drain on that account balance. For example, a startup with \$1 million in the bank that is spending \$100,000 per month has a projected runway of ten months.

When startups start to run low on cash, they can extend the runway two ways: by cutting costs or by raising additional funds. But when entrepreneurs cut costs indiscriminately, they are as liable to cut the costs that are allowing the company to get through its Build-Measure-Learn feedback loop as they are to cut waste. If the cuts result in a slowdown to this feedback loop, all they have accomplished is to help the startup go out of business more slowly.

The true measure of runway is how many pivots a startup has left: the number of opportunities it has to make a fundamental change to its business strategy. Measuring runway through the lens of pivots rather than that of time suggests another way to extend that runway: get to each pivot faster. In other words, the startup has to find ways to achieve the same amount of validated learning at lower cost or in a shorter time. All the techniques in the Lean Startup model that have been discussed so far have this as their overarching goal.

## PIVOTS REQUIRE COURAGE

Ask most entrepreneurs who have decided to pivot and they will tell you that they wish they had made the decision sooner. I believe there are three reasons why this happens.

First, vanity metrics can allow entrepreneurs to form false conclusions and live in their own private reality. This is particularly damaging to the decision to pivot because it robs teams of the belief that it is necessary to change. When people are forced to change against their better judgment, the process is harder, takes longer, and leads to a less decisive outcome.

Second, when an entrepreneur has an unclear hypothesis, it's almost impossible to experience complete failure, and without failure there is usually no impetus to embark on the radical change a pivot requires. As I mentioned earlier, the failure of the “launch it and see what happens” approach should now be evident: you will always succeed—in seeing what happens. Except in rare cases, the early results will be ambiguous, and you won't know whether to

pivot or persevere, whether to change direction or stay the course.

Third, many entrepreneurs are afraid. Acknowledging failure can lead to dangerously low morale. Most entrepreneurs' biggest fear is not that their vision will prove to be wrong. More terrifying is the thought that the vision might be deemed wrong without having been given a real chance to prove itself. This fear drives much of the resistance to the minimum viable product, split testing, and other techniques to test hypotheses. Ironically, this fear drives up the risk because testing doesn't occur until the vision is fully represented. However, by that time it is often too late to pivot because funding is running out. To avoid this fate, entrepreneurs need to face their fears and be willing to fail, often in a public way. In fact, entrepreneurs who have a high profile, either because of personal fame or because they are operating as part of a famous brand, face an extreme version of this problem.

A new startup in Silicon Valley called Path was started by experienced entrepreneurs: Dave Morin, who previously had overseen Facebook's platform initiative; Dustin Mierau, product designer and cocreator of Macster; and Shawn Fanning of Napster fame. They decided to release a minimum viable product in 2010. Because of the high-profile nature of its founders, the MVP attracted significant press attention, especially from technology and startup blogs. Unfortunately, their product was not targeted at technology early adopters, and as a result, the early blogger reaction was quite negative. (Many entrepreneurs fail to launch because they are afraid of this kind of reaction, worrying that it will harm the morale of the entire company. The allure of positive press, especially in our "home" industry, is quite strong.)

Luckily, the Path team had the courage to ignore this fear and focus on what their customers said. As a result, they were able to get essential early feedback from actual customers. Path's goal is to create a more personal social network that maintains its quality over time. Many people have had the experience of being overconnected on existing social networks, sharing with past coworkers, high school friends, relatives, and colleagues. Such broad groups make it hard to share intimate moments. Path took an

unusual approach. For example, it limited the number of connections to fifty, based on brain research by the anthropologist Robin Dunbar at Oxford. His research suggests that fifty is roughly the number of personal relationships in any person's life at any given time.

For members of the tech press (and many tech early adopters) this “artificial” constraint on the number of connections was anathema. They routinely use new social networking products with thousands of connections. Fifty seemed way too small. As a result, Path endured a lot of public criticism, which was hard to ignore. But customers flocked to the platform, and their feedback was decidedly different from the negativity in the press. Customers liked the intimate moments and consistently wanted features that were not on the original product road map, such as the ability to share how friends' pictures made them feel and the ability to share “video moments.”

Dave Morin summed up his experience this way:

The reality of our team and our backgrounds built up a massive wall of expectations. I don't think it would have mattered what we would have released; we would have been met with expectations that are hard to live up to. But to us it just meant we needed to get our product and our vision out into the market broadly in order to get feedback and to begin iteration. We humbly test our theories and our approach to see what the market thinks. Listen to feedback honestly. And continue to innovate in the directions we think will create meaning in the world.

Path's story is just beginning, but already their courage in facing down critics is paying off. If and when they need to pivot, they won't be hampered by fear. They recently raised \$8.5 million in venture capital in a round led by Kleiner Perkins Caufield & Byers. In doing so, Path reportedly turned down an acquisition offer for \$100 million from Google.<sup>2</sup>

## THE PIVOT OR PERSEVERE MEETING

The decision to pivot requires a clear-eyed and objective mind-set. We've discussed the telltale signs of the need to pivot: the decreasing effectiveness of product experiments and the general feeling that product development should be more productive. Whenever you see those symptoms, consider a pivot.

The decision to pivot is emotionally charged for any startup and has to be addressed in a structured way. One way to mitigate this challenge is to schedule the meeting in advance. I recommend that every startup have a regular "pivot or persevere" meeting. In my experience, less than a few weeks between meetings is too often and more than a few months is too infrequent. However, each startup needs to find its own pace.

Each pivot or persevere meeting requires the participation of both the product development and business leadership teams. At IMVU, we also added the perspectives of outside advisers who could help us see past our preconceptions and interpret data in new ways. The product development team must bring a complete report of the results of its product optimization efforts over time (not just the past period) as well as a comparison of how those results stack up against expectations (again, over time). The business leadership should bring detailed accounts of their conversations with current and potential customers.

Let's take a look at this process in action in a dramatic pivot done by a company called Wealthfront. That company was founded in 2007 by Dan Carroll and added Andy Rachleff as CEO shortly thereafter. Andy is a well-known figure in Silicon Valley: he is a cofounder and former general partner of the venture capital firm Benchmark Capital and is on the faculty of the Stanford Graduate School of Business, where he teaches a variety of courses on technology entrepreneurship. I first met Andy when he commissioned a case study on IMVU to teach his students about the process we had used to build the company.

Wealthfront's mission is to disrupt the mutual fund industry by bringing greater transparency, access, and value to retail investors.

What makes Wealthfront's story unusual, however, is not where it is today but how it began: as an online game.

In Wealthfront's original incarnation it was called kaChing and was conceived as a kind of fantasy league for amateur investors. It allowed anyone to open a virtual trading account and build a portfolio that was based on real market data without having to invest real money. The idea was to identify diamonds in the rough: amateur traders who lacked the resources to become fund managers but who possessed market insight. Wealthfront's founders did not want to be in the online gaming business per se; kaChing was part of a sophisticated strategy in the service of their larger vision. Any student of disruptive innovation would have looked on approvingly: they were following that system perfectly by initially serving customers who were unable to participate in the mainstream market. Over time, they believed, the product would become more and more sophisticated, eventually allowing users to serve (and disrupt) existing professional fund managers.

To identify the best amateur trading savants, Wealthfront built sophisticated technology to rate the skill of each fund manager, using techniques employed by the most sophisticated evaluators of money managers, the premier U.S. university endowments. Those methods allowed them to evaluate not just the returns the managers generated but also the amount of risk they had taken along with how consistent they performed relative to their declared investment strategy. Thus, fund managers who achieved great returns through reckless gambles (i.e., investments outside their area of expertise) would be ranked lower than those who had figured out how to beat the market through skill.

With its kaChing game, Wealthfront hoped to test two leap-of-faith assumptions:

1. A significant percentage of the game players would demonstrate enough talent as virtual fund managers to prove themselves suitable to become managers of real assets (the value hypothesis).



2. The game would grow using the viral engine of growth and generate value using a freemium business model. The game was free to play, but the team hoped that a percentage of the players would realize that they were lousy traders and therefore want to convert to paying customers once Wealthfront started offering real asset management services (the growth hypothesis).

kaChing was a huge early success, attracting more than 450,000 gamers in its initial launch. By now, you should be suspicious of this kind of vanity metric. Many less disciplined companies would have celebrated that success and felt their future was secure, but Wealthfront had identified its assumptions clearly and was able to think more rigorously. By the time Wealthfront was ready to launch its paid financial product, only seven amateur managers had qualified as worthy of managing other people's money, far less than the ideal model had anticipated. After the paid product launched, they were able to measure the conversion rate of gamers into paying customers. Here too the numbers were discouraging: the conversion rate was close to zero. Their model had predicted that hundreds of customers would sign up, but only fourteen did.

The team worked valiantly to find ways to improve the product, but none showed any particular promise. It was time for a pivot or persevere meeting.

If the data we have discussed so far was all that was available at that critical meeting, Wealthfront would have been in trouble. They would have known that their current strategy wasn't working but not what to do to fix it. That is why it was critical that they followed the recommendation earlier in this chapter to investigate alternative possibilities. In this case, Wealthfront had pursued two important lines of inquiry.

The first was a series of conversations with professional money managers, beginning with John Powers, the head of Stanford University's endowment, who reacted surprisingly positively. Wealthfront's strategy was premised on the assumption that professional money managers would be reluctant to join the system

because the increased transparency would threaten their sense of authority. Powers had no such concerns. CEO Andy Rachleff then began a series of conversations with other professional investment managers and brought the results back to the company. His insights were as follows:

1. Successful professional money managers felt they had nothing to fear from transparency, since they believed it would validate their skills.
2. Money managers faced significant challenges in managing and scaling their own businesses. They were hampered by the difficulty of servicing their own accounts and therefore had to require high minimum investments as a way to screen new clients.

The second problem was so severe that Wealthfront was fielding cold calls from professional managers asking out of the blue to join the platform. These were classic early adopters who had the vision to see past the current product to something they could use to achieve a competitive advantage.

The second critical qualitative information came out of conversations with consumers. It turned out that they found the blending of virtual and real portfolio management on the kaChing website confusing. Far from being a clever way of acquiring customers, the freemium strategy was getting in the way by promoting confusion about the company's positioning.

This data informed the pivot or persevere meeting. With everyone present, the team debated what to do with its future. The current strategy wasn't working, but many employees were nervous about abandoning the online game. After all, it was an important part of what they had signed on to build. They had invested significant time and energy building and supporting those customers. It was painful—as it always is—to realize that that energy had been wasted.

Wealthfront decided it could not persevere as it existed. The company chose instead to celebrate what it had learned. If it had

not launched its current product, the team never would have learned what it needed to know to pivot. In fact, the experience taught them something essential about their vision. As Andy says, “What we really wanted to change was not who manages the money but who has access to the best possible talent. We’d originally thought we’d need to build a significant business with amateur managers to get professionals to come on board, but fortunately it turns out that wasn’t necessary.”

The company pivoted, abandoning the gaming customers altogether and focusing on providing a service that allowed customers to invest with professional managers. On the surface, the pivot seems quite dramatic in that the company changed its positioning, its name, and its partner strategy. It even jettisoned a large proportion of the features it had built. But at its core, a surprising amount stayed the same. The most valuable work the company had done was building technology to evaluate managers’ effectiveness, and this became the kernel around which the new business was built. This is also common with pivots; it is not necessary to throw out everything that came before and start over. Instead, it’s about repurposing what has been built and what has been learned to find a more positive direction.

Today, Wealthfront is prospering as a result of its pivot, with over \$180 million invested on the platform and more than forty professional managers.<sup>3</sup> It recently was named one of Fast Company’s ten most innovative companies in finance.<sup>4</sup> The company continues to operate with agility, scaling in line with the growth principles outlined in [Chapter 12](#). Wealthfront is also a leading advocate of the development technique known as continuous deployment, which we’ll discuss in [Chapter 9](#).

## FAILURE TO PIVOT

The decision to pivot is so difficult that many companies fail to make it. I wish I could say that every time I was confronted with the need to pivot, I handled it well, but this is far from true. I

remember one failure to pivot especially well.

A few years after IMVU's founding, the company was having tremendous success. The business had grown to over \$1 million per month in revenue; we had created more than twenty million avatars for our customers. We managed to raise significant new rounds of financing, and like the global economy, we were riding high. But danger lurked around the corner.

Unknowingly, we had fallen into a classic startup trap. We had been so successful with our early efforts that we were ignoring the principles behind them. As a result, we missed the need to pivot even as it stared us in the face.

We had built an organization that excelled at the kinds of activities described in earlier chapters: creating minimum viable products to test new ideas and running experiments to tune the engine of growth. Before we had begun to enjoy success, many people had advised against our "low-quality" minimum viable product and experimental approach, urging us to slow down. They wanted us to do things right and focus on quality instead of speed. We ignored that advice, mostly because we wanted to claim the advantages of speed. After our approach was vindicated, the advice we received changed. Now most of the advice we heard was that "you can't argue with success," urging us to stay the course. We liked this advice better, but it was equally wrong.

Remember that the rationale for building low-quality MVPs is that developing any features beyond what early adopters require is a form of waste. However, the logic of this takes you only so far. Once you have found success with early adopters, you want to sell to mainstream customers. Mainstream customers have different requirements and are much more demanding.

The kind of pivot we needed is called a customer segment pivot. In this pivot, the company realizes that the product it's building solves a real problem for real customers but that they are not the customers it originally planned to serve. In other words, the product hypothesis is confirmed only partially. (This chapter described such a pivot in the Votizen story, above.)

A customer segment pivot is an especially tricky pivot to execute

because, as we learned the hard way at IMVU, the very actions that made us successful with early adopters were diametrically opposed to the actions we'd have to master to be successful with mainstream customers. We lacked a clear understanding of how our engine of growth operated. We had begun to trust our vanity metrics. We had stopped using learning milestones to hold ourselves accountable. Instead, it was much more convenient to focus on the ever-larger gross metrics that were so exciting: breaking new records in signing up paying customers and active users, monitoring our customer retention rate—you name it. Under the surface, it should have been clear that our efforts at tuning the engine were reaching diminishing returns, the classic sign of the need to pivot.

For example, we spent months trying to improve the product's activation rate (the rate at which new customers become active consumers of the product), which remained stubbornly low. We did countless experiments: usability improvements, new persuasion techniques, incentive programs, customer quests, and other game-like features. Individually, many of these new features and new marketing tools were successful. We measured them rigorously, using A/B experimentation. But taken in aggregate, over the course of many months, we were seeing negligible changes in the overall drivers of our engine of growth. Even our activation rate, which had been the center of our focus, edged up only a few percentage points.

We ignored the signs because the company was still growing, delivering month after month of “up and to the right” results. But we were quickly exhausting our early adopter market. It was getting harder and harder to find customers we could acquire at the prices we were accustomed to paying. As we drove our marketing team to find more customers, they were forced to reach out more to mainstream customers, but mainstream customers are less forgiving of an early product. The activation and monetization rates of new customers started to go down, driving up the cost of acquiring new customers. Pretty soon, our growth was flatlining and our engine sputtered and stalled.

It took us far too long to make the changes necessary to fix this

situation. As with all pivots, we had to get back to basics and start the innovation accounting cycle over. It felt like the company's second founding. We had gotten really good at optimizing, tuning, and iterating, but in the process we had lost sight of the purpose of those activities: testing a clear hypothesis in the service of the company's vision. Instead, we were chasing growth, revenue, and profits wherever we could find them.

We needed to reacquaint ourselves with our new mainstream customers. Our interaction designers led the way by developing a clear customer archetype that was based on extensive in-person conversations and observation. Next, we needed to invest heavily in a major product overhaul designed to make the product dramatically easier to use. Because of our overfocus on fine-tuning, we had stopped making large investments like these, preferring to invest in lower-risk and lower-yield testing experiments.

However, investing in quality, design, and larger projects did not require that we abandon our experimental roots. On the contrary, once we realized our mistake and executed the pivot, those skills served us well. We created a sandbox for experimentation like the one described in [Chapter 12](#) and had a cross-functional team work exclusively on this major redesign. As they built, they continuously tested their new design head to head against the old one. Initially, the new design performed worse than the old one, as is usually the case. It lacked the features and functionality of the old design and had many new mistakes as well. But the team relentlessly improved the design until, months later, it performed better. This new design laid the foundation for our future growth.

This foundation has paid off handsomely. By 2009, revenue had more than doubled to over \$25 million annually. But we might have enjoyed that success earlier if we had pivoted sooner.<sup>5</sup>

## A CATALOG OF PIVOTS

Pivots come in different flavors. The word pivot sometimes is used incorrectly as a synonym for change. A pivot is a special kind of

change designed to test a new fundamental hypothesis about the product, business model, and engine of growth.

### Zoom-in Pivot

In this case, what previously was considered a single feature in a product becomes the whole product. This is the type of pivot Votizen made when it pivoted away from a full social network and toward a simple voter contact product.

### Zoom-out Pivot

In the reverse situation, sometimes a single feature is insufficient to support a whole product. In this type of pivot, what was considered the whole product becomes a single feature of a much larger product.

### Customer Segment Pivot

In this pivot, the company realizes that the product it is building solves a real problem for real customers but that they are not the type of customers it originally planned to serve. In other words, the product hypothesis is partially confirmed, solving the right problem, but for a different customer than originally anticipated.

### Customer Need Pivot

As a result of getting to know customers extremely well, it sometimes becomes clear that the problem we're trying to solve for them is not very important. However, because of this customer intimacy, we often discover other related problems that are important and can be solved by our team. In many cases, these related problems may require little more than repositioning the

existing product. In other cases, it may require a completely new product. Again, this a case where the product hypothesis is partially confirmed; the target customer has a problem worth solving, just not the one that was originally anticipated.

A famous example is the chain Potbelly Sandwich Shop, which today has over two hundred stores. It began as an antique store in 1977; the owners started to sell sandwiches as a way to bolster traffic to their stores. Pretty soon they had pivoted their way into an entirely different line of business.

## Platform Pivot

A platform pivot refers to a change from an application to a platform or vice versa. Most commonly, startups that aspire to create a new platform begin life by selling a single application, the so-called killer app, for their platform. Only later does the platform emerge as a vehicle for third parties to leverage as a way to create their own related products. However, this order is not always set in stone, and some companies have to execute this pivot multiple times.

## Business Architecture Pivot

This pivot borrows a concept from Geoffrey Moore, who observed that companies generally follow one of two major business architectures: high margin, low volume (complex systems model) or low margin, high volume (volume operations model).<sup>6</sup> The former commonly is associated with business to business (B2B) or enterprise sales cycles, and the latter with consumer products (there are notable exceptions). In a business architecture pivot, a startup switches architectures. Some companies change from high margin, low volume by going mass market (e.g., Google's search "appliance"); others, originally designed for the mass market, turned out to require long and expensive sales cycles.



## Value Capture Pivot

There are many ways to capture the value a company creates. These methods are referred to commonly as monetization or revenue models. These terms are much too limiting. Implicit in the idea of monetization is that it is a separate “feature” of a product that can be added or removed at will. In reality, capturing value is an intrinsic part of the product hypothesis. Often, changes to the way a company captures value can have far-reaching consequences for the rest of the business, product, and marketing strategies.

## Engine of Growth Pivot

As we’ll see in [Chapter 10](#), there are three primary engines of growth that power startups: the viral, sticky, and paid growth models. In this type of pivot, a company changes its growth strategy to seek faster or more profitable growth. Commonly but not always, the engine of growth also requires a change in the way value is captured.

## Channel Pivot

In traditional sales terminology, the mechanism by which a company delivers its product to customers is called the sales channel or distribution channel. For example, consumer packaged goods are sold in a grocery store, cars are sold in dealerships, and much enterprise software is sold (with extensive customization) by consulting and professional services firms. Often, the requirements of the channel determine the price, features, and competitive landscape of a product. A channel pivot is a recognition that the same basic solution could be delivered through a different channel with greater effectiveness. Whenever a company abandons a previously complex sales process to “sell direct” to its end users, a channel pivot is in progress.

It is precisely because of its destructive effect on sales channels that the Internet has had such a disruptive influence in industries that previously required complex sales and distribution channels, such as newspaper, magazine, and book publishing.

## Technology Pivot

Occasionally, a company discovers a way to achieve the same solution by using a completely different technology. Technology pivots are much more common in established businesses. In other words, they are a sustaining innovation, an incremental improvement designed to appeal to and retain an existing customer base. Established companies excel at this kind of pivot because so much is not changing. The customer segment is the same, the customer's problem is the same, the value-capture model is the same, and the channel partners are the same. The only question is whether the new technology can provide superior price and/or performance compared with the existing technology.

## A PIVOT IS A STRATEGIC HYPOTHESIS

Although the pivots identified above will be familiar to students of business strategy, the ability to pivot is no substitute for sound strategic thinking. The problem with providing famous examples of pivots is that most people are familiar only with the successful end strategies of famous companies. Most readers know that Southwest or Walmart is an example of a low-cost disruption in their markets, that Microsoft an example of a platform monopoly, and that Starbucks has leveraged a powerful premium brand. What is generally less well known are the pivots that were required to discover those strategies. Companies have a strong incentive to align their PR stories around the heroic founder and make it seem that their success was the inevitable result of a good idea.

Thus, although startups often pivot into a strategy that seems

similar to that of a successful company, it is important not to put too much stock in these analogies. It's extremely difficult to know if the analogy has been drawn properly. Have we copied the essential features or just superficial ones? Will what worked in that industry work in ours? Will what has worked in the past work today? A pivot is better understood as a new strategic hypothesis that will require a new minimum viable product to test.

Pivots are a permanent fact of life for any growing business. Even after a company achieves initial success, it must continue to pivot. Those familiar with the technology life cycle ideas of theorists such as Geoffrey Moore know certain later-stage pivots by the names he has given them: the Chasm, the Tornado, the Bowling Alley. Readers of the disruptive innovation literature spearheaded by Harvard's Clayton Christensen will be familiar with established companies that fail to pivot when they should. The critical skill for managers today is to match those theories to their present situation so that they apply the right advice at the right time.

Modern managers cannot have escaped the deluge of recent books calling on them to adapt, change, reinvent, or upend their existing businesses. Many of the works in this category are long on exhortations and short on specifics.

A pivot is not just an exhortation to change. Remember, it is a special kind of structured change designed to test a new fundamental hypothesis about the product, business model, and engine of growth. It is the heart of the Lean Startup method. It is what makes the companies that follow Lean Startup resilient in the face of mistakes: if we take a wrong turn, we have the tools we need to realize it and the agility to find another path.



In [Part Two](#), we have looked at a startup idea from its initial leaps of faith, tested it with a minimum viable product, used innovation accounting and actionable metrics to evaluate the results, and made the decision to pivot or persevere.

I have treated these subjects in great detail to prepare for what

comes next. On the page, these processes may seem clinical, slow, and simple. In the real world, something different is needed. We have learned to steer when moving slowly. Now we must learn to race. Laying a solid foundation is only the first step toward our real destination: acceleration.



# Part Three

## ACCELERATE

# Start Your Engines

Most of the decisions startups face are not clear-cut. How often should you release a product? Is there a reason to release weekly rather than daily or quarterly or annually? Product releases incur overhead, and so from an efficiency point of view, releasing often leaves less time to devote to building the product. However, waiting too long to release can lead to the ultimate waste: making something that nobody wants.

How much time and energy should companies invest in infrastructure and planning early on in anticipation of success? Spend too much and you waste precious time that could have been spent learning. Spend too little and you may fail to take advantage of early success and cede market leadership to a fast follower.

What should employees spend their days doing? How do we hold people accountable for learning at an organizational level? Traditional departments create incentive structures that keep people focused on excellence in their specialties: marketing, sales, product development. But what if the company's best interests are served by cross-functional collaboration? Startups need organizational structures that combat the extreme uncertainty that is a startup's chief enemy.

The lean manufacturing movement faced similar questions on the factory floor. Their answers are relevant for startups as well, with some modifications.

The critical first question for any lean transformation is: which

activities create value and which are a form of waste? Once you understand this distinction, you can begin using lean techniques to drive out waste and increase the efficiency of the value-creating activities. For these techniques to be used in a startup, they must be adapted to the unique circumstances of entrepreneurship. Recall from [Chapter 3](#) that value in a startup is not the creation of stuff, but rather validated learning about how to build a sustainable business. What products do customers really want? How will our business grow? Who is our customer? Which customers should we listen to and which should we ignore? These are the questions that need answering as quickly as possible to maximize a startup's chances of success. That is what creates value for a startup.

In [Part Three](#), we will develop techniques that allow Lean Startups to grow without sacrificing the speed and agility that are the lifeblood of every startup. Contrary to common belief, lethargy and bureaucracy are not the inevitable fate of companies as they achieve maturity. I believe that with the proper foundation, Lean Startups can grow to become lean enterprises that maintain their agility, learning orientation, and culture of innovation even as they scale.

In [Chapter 9](#), we will see how Lean Startups take advantage of the counterintuitive power of small batches. Just as lean manufacturing has pursued a just-in-time approach to building products, reducing the need for in-process inventory, Lean Startups practice just-in-time scalability, conducting product experiments without making massive up-front investments in planning and design.

[Chapter 10](#) will explore the metrics startups should use to understand their growth as they add new customers and discover new markets. Sustainable growth follows one of three engines of growth: paid, viral, or sticky. By identifying which engine of growth a startup is using, it can then direct energy where it will be most effective in growing the business. Each engine requires a focus on unique metrics to evaluate the success of new products and prioritize new experiments. When used with the innovation accounting method described in [Part Two](#), these metrics allow



startups to figure out when their growth is at risk of running out and pivot accordingly.

**Chapter 11** shows how to build an adaptive organization by investing in the right amount of process to keep teams nimble as they grow. We will see how techniques from the tool kit of lean manufacturing, such as the Five Whys, help startup teams grow without becoming bureaucratic or dysfunctional. We also will see how lean disciplines set the stage for a startup to transition into an established company driven by operational excellence.

In **Chapter 12**, we'll come full circle. As startups grow into established companies, they face the same pressures that make it necessary for today's enterprises to find new ways to invest in disruptive innovation. In fact, we'll see that an advantage of a successful startup's rapid growth is that the company can keep its entrepreneurial DNA even as it matures. Today's companies must learn to master a management portfolio of sustainable and disruptive innovation. It is an obsolete view that sees startups as going through discrete phases that leave earlier kinds of work—such as innovation—behind. Rather, modern companies must excel at doing multiple kinds of work in parallel. To do so, we'll explore techniques for incubating innovation teams within the context of an established company.

I have included an epilogue called “Waste Not” in which I consider some of the broader implications of the success of the Lean Startup movement, place it in historical context (including cautionary lessons from past movements), and make suggestions for its future direction.

## BATCH

In the book *Lean Thinking*, James Womack and Daniel Jones recount a story of stuffing newsletters into envelopes with the assistance of one of the author's two young children. Every envelope had to be addressed, stamped, filled with a letter, and sealed. The daughters, age six and nine, knew how they should go about completing the project: "Daddy, first you should fold all of the newsletters. Then you should attach the seal. Then you should put on the stamps." Their father wanted to do it the counterintuitive way: complete each envelope one at a time. They—like most of us—thought that was backward, explaining to him "that wouldn't be efficient!" He and his daughters each took half the envelopes and competed to see who would finish first.

The father won the race, and not just because he is an adult. It happened because the one envelope at a time approach is a faster way of getting the job done even though it seems inefficient. This has been confirmed in many studies, including one that was recorded on video.<sup>1</sup>

The one envelope at a time approach is called "single-piece flow" in lean manufacturing. It works because of the surprising power of small batches. When we do work that proceeds in stages, the "batch size" refers to how much work moves from one stage to the next at a time. For example, if we were stuffing one hundred envelopes, the intuitive way to do it—folding one hundred letters at a time—would have a batch size of one hundred. Single-piece flow is so named because it has a batch size of one.

Why does stuffing one envelope at a time get the job done faster even though it seems like it would be slower? Because our intuition doesn't take into account the extra time required to sort, stack, and move around the large piles of half-complete envelopes when it's done the other way.<sup>2</sup> It seems more efficient to repeat the same task over and over, in part because we expect that we will get better at this simple task the more we do it. Unfortunately, in process-oriented work like this, individual performance is not nearly as important as the overall performance of the system.

Even if the amount of time that each process took was exactly the same, the small batch production approach still would be superior, and for even more counterintuitive reasons. For example, imagine that the letters didn't fit in the envelopes. With the large-batch approach, we wouldn't find that out until nearly the end. With small batches, we'd know almost immediately. What if the envelopes are defective and won't seal? In the large-batch approach, we'd have to unstuff all the envelopes, get new ones, and restuff them. In the small-batch approach, we'd find this out immediately and have no rework required.

All these issues are visible in a process as simple as stuffing envelopes, but they are of real and much greater consequence in the work of every company, large or small. The small-batch approach produces a finished product every few seconds, whereas the large-batch approach must deliver all the products at once, at the end. Imagine what this might look like if the time horizon was hours, days, or weeks. What if it turns out that the customers have decided they don't want the product? Which process would allow a company to find this out sooner?

Lean manufacturers discovered the benefits of small batches decades ago. In the post-World War II economy, Japanese carmakers such as Toyota could not compete with huge American factories that used the latest mass production techniques. Following the intuitively efficient way of building, mass production factories built cars by using ever-larger batch sizes. They would spend huge amounts of money buying machines that could produce car parts by

the tens, hundreds, or thousands. By keeping those machines running at peak speed, they could drive down the unit cost of each part and produce cars that were incredibly inexpensive so long as they were completely uniform.

The Japanese car market was far too small for companies such as Toyota to employ those economies of scale; thus, Japanese companies faced intense pressure from mass production. Also, in the war-ravaged Japanese economy, capital was not available for massive investments in large machines.

It was against this backdrop that innovators such as Taiichi Ohno, Shigeo Shingo, and others found a way to succeed by using small batches. Instead of buying large specialized machines that could produce thousands of parts at a time, Toyota used smaller general-purpose machines that could produce a wide variety of parts in small batches. This required figuring out ways to reconfigure each machine rapidly to make the right part at the right time. By focusing on this “changeover time,” Toyota was able to produce entire automobiles by using small batches throughout the process.

This rapid changing of machines was no easy feat. As in any lean transformation, existing systems and tools often need to be reinvented to support working in smaller batches. Shigeo Shingo created the concept of SMED (Single-Minute Exchange of Die) in order to enable a smaller batch size of work in early Toyota factories. He was so relentless in rethinking the way machines were operated that he was able to reduce changeover times that previously took hours to less than ten minutes. He did this, not by asking workers to work faster, but by reimagining and restructuring the work that needed to be done. Every investment in better tools and process had a corresponding benefit in terms of shrinking the batch size of work.

Because of its smaller batch size, Toyota was able to produce a much greater diversity of products. It was no longer necessary that each product be exactly the same to gain the economies of scale that powered mass production. Thus, Toyota could serve its smaller, more fragmented markets and still compete with the mass producers. Over time, that capability allowed Toyota to move

successfully into larger and larger markets until it became the world's largest automaker in 2008.

The biggest advantage of working in small batches is that quality problems can be identified much sooner. This is the origin of Toyota's famous andon cord, which allows any worker to ask for help as soon as they notice any problem, such as a defect in a physical part, stopping the entire production line if it cannot be corrected immediately. This is another very counterintuitive practice. An assembly line works best when it is functioning smoothly, rolling car after car off the end of the line. The andon cord can interrupt this careful flow as the line is halted repeatedly. However, the benefits of finding and fixing problems faster outweigh this cost. This process of continuously driving out defects has been a win-win for Toyota and its customers. It is the root cause of Toyota's historic high quality ratings and low costs.

## SMALL BATCHES IN ENTREPRENEURSHIP

When I teach entrepreneurs this method, I often begin with stories about manufacturing. Before long, I can see the questioning looks: what does this have to do with my startup? The theory that is the foundation of Toyota's success can be used to dramatically improve the speed at which startups find validated learning.

Toyota discovered that small batches made their factories more efficient. In contrast, in the Lean Startup the goal is not to produce more stuff efficiently. It is to—as quickly as possible—learn how to build a sustainable business.

Think back to the example of envelope stuffing. What if it turns out that the customer doesn't want the product we're building? Although this is never good news for an entrepreneur, finding out sooner is much better than finding out later. Working in small batches ensures that a startup can minimize the expenditure of time, money, and effort that ultimately turns out to have been wasted.

## Small Batches at IMVU

At IMVU, we applied these lessons from manufacturing to the way we work. Normally, new versions of products like ours are released to customers on a monthly, quarterly, or yearly cycle.

Take a look at your cell phone. Odds are, it is not the very first version of its kind. Even innovative companies such as Apple produce a new version of their flagship phones about once a year. Bundled up in that product release are dozens of new features (at the release of iPhone 4, Apple boasted more than 1,500 changes).

Ironically, many high-tech products are manufactured in advanced facilities that follow the latest in lean thinking, including small batches and single-piece flow. However, the process that is used to design the product is stuck in the era of mass production. Think of all the changes that are made to a product such as the iPhone; all 1,500 of them are released to customers in one giant batch.

Behind the scenes, in the development and design of the product itself, large batches are still the rule. The work that goes into the development of a new product proceeds on a virtual assembly line. Product managers figure out what features are likely to please customers; product designers then figure out how those features should look and feel. These designs are passed to engineering, which builds something new or modifies an existing product and, once this is done, hands it off to somebody responsible for verifying that the new product works the way the product managers and designers intended. For a product such as the iPhone, these internal handoffs may happen on a monthly or quarterly basis.

Think back one more time to the envelope-stuffing exercise. What is the most efficient way to do this work?

At IMVU, we attempted to design, develop, and ship our new features one at a time, taking advantage of the power of small batches. Here's what it looked like.

Instead of working in separate departments, engineers and designers would work together side by side on one feature at a time. Whenever that feature was ready to be tested with customers,

they immediately would release a new version of the product, which would go live on our website for a relatively small number of people. The team would be able immediately to assess the impact of their work, evaluate its effect on customers, and decide what to do next. For tiny changes, the whole process might be repeated several times per day. In fact, in the aggregate, IMVU makes about fifty changes to its product (on average) every single day.

Just as with the Toyota Production System, the key to being able to operate this quickly is to check for defects immediately, thus preventing bigger problems later. For example, we had an extensive set of automated tests that assured that after every change our product still worked as designed. Let's say an engineer accidentally removed an important feature, such as the checkout button on one of our e-commerce pages. Without this button, customers no longer could buy anything from IMVU. It's as if our business instantly became a hobby. Analogously to the Toyota andon cord, IMVU used an elaborate set of defense mechanisms that prevented engineers from accidentally breaking something important.

We called this our product's immune system because those automatic protections went beyond checking that the product behaved as expected. We also continuously monitored the health of our business itself so that mistakes were found and removed automatically.

Going back to our business-to-hobby example of the missing checkout button, let's make the problem a little more interesting. Imagine that instead of removing the button altogether, an engineer makes a mistake and changes the button's color so that it is now white on a white background. From the point of view of automated functional tests, the button is still there and everything is working normally; from the customer's point of view, the button is gone, and so nobody can buy anything. This class of problems is hard to detect solely with automation but is still catastrophic from a business point of view. At IMVU, our immune system is programmed to detect these business consequences and

automatically invoke our equivalent of the andon cord.

When our immune system detects a problem, a number of things happen immediately:

1. The defective change is removed immediately and automatically.
2. Everyone on the relevant team is notified of the problem.
3. The team is blocked from introducing any further changes, preventing the problem from being compounded by future mistakes ...
4. ... until the root cause of the problem is found and fixed. (This root cause analysis is discussed in greater detail in [Chapter 11](#).)

At IMVU, we called this continuous deployment, and even in the fast-moving world of software development it is still considered controversial.<sup>3</sup> As the Lean Startup movement has gained traction, it has come to be embraced by more and more startups, even those that operate mission-critical applications. Among the most cutting edge examples is Wealthfront, whose pivot was described in [Chapter 8](#). The company practices true continuous deployment—including more than a dozen releases to customers every day—in an SEC-regulated environment.<sup>4</sup>

## Continuous Deployment Beyond Software

When I tell this story to people who work in a slower-moving industry, they think I am describing something futuristic. But increasingly, more and more industries are seeing their design process accelerated by the same underlying forces that make this kind of rapid iteration possible in the software industry. There are three ways in which this is happening:

1. Hardware becoming software. Think about what has happened in consumer electronics. The latest phones and tablet computers are



little more than a screen connected to the Internet. Almost all of their value is determined by their software. Even old-school products such as automobiles are seeing ever-larger parts of their value being generated by the software they carry inside, which controls everything from the entertainment system to tuning the engine to controlling the brakes. What can be built out of software can be modified much faster than a physical or mechanical device can.

2. Fast production changes. Because of the success of the lean manufacturing movement, many assembly lines are set up to allow each new product that comes off the line to be customized completely without sacrificing quality or cost-effectiveness. Historically, this has been used to offer the customer many choices of product, but in the future, this capability will allow the designers of products to get much faster feedback about new versions. When the design changes, there is no excess inventory of the old version to slow things down. Since machines are designed for rapid changeovers, as soon as the new design is ready, new versions can be produced quickly.

3. 3D printing and rapid prototyping tools. As just one example, most products and parts that are made out of plastic today are mass produced using a technique called injection molding. This process is extremely expensive and time-consuming to set up, but once it is up and running, it can reproduce hundreds of thousands of identical individual items at an extremely low cost. It is a classic large-batch production process. This has put entrepreneurs who want to develop a new physical product at a disadvantage, since in general only large companies can afford these large production runs for a new product. However, new technologies are allowing entrepreneurs to build small batches of products that are of the same quality as products made with injection molding, but at much lower cost and much, much faster.

The essential lesson is not that everyone should be shipping fifty times per day but that by reducing batch size, we can get through the Build-Measure-Learn feedback loop more quickly than our competitors can. The ability to learn faster from customers is the essential competitive advantage that startups must possess.

## SMALL BATCHES IN ACTION

To see this process in action, let me introduce you to a company in Boise, Idaho, called SGW Designworks. SGW's specialty is rapid production techniques for physical products. Many of its clients are startups.

SGW Designworks was engaged by a client who had been asked by a military customer to build a complex field x-ray system to detect explosives and other destructive devices at border crossings and in war zones.

Conceptually, the system consisted of an advanced head unit that read x-ray film, multiple x-ray film panels, and the framework to hold the panels while the film was being exposed. The client already had the technology for the x-ray panels and the head unit, but to make the product work in rugged military settings, SGW needed to design and deliver the supporting structure that would make the technology usable in the field. The framework had to be stable to ensure a quality x-ray image, durable enough for use in a war zone, easy to deploy with minimal training, and small enough to collapse into a backpack.

This is precisely the kind of product we are accustomed to thinking takes months or years to develop, yet new techniques are shrinking that time line. SGW immediately began to generate the visual prototypes by using 3D computer-aided design (CAD) software. The 3D models served as a rapid communication tool between the client and the SGW team to make early design decisions.

The team and client settled on a design that used an advanced

locking hinge to provide the collapsibility required without compromising stability. The design also integrated a suction cup/pump mechanism to allow for fast, repeatable attachment to the x-ray panels. Sounds complicated, right?

Three days later, the SGW team delivered the first physical prototypes to the client. The prototypes were machined out of aluminum directly from the 3D model, using a technique called computer numerical control (CNC) and were hand assembled by the SGW team.

The client immediately took the prototypes to its military contact for review. The general concept was accepted with a number of minor design modifications. In the next five days, another full cycle of design iteration, prototyping, and design review was completed by the client and SGW. The first production run of forty completed units was ready for delivery three and a half weeks after the initiation of the development project.

SGW realized that this was a winning model because feedback on design decisions was nearly instantaneous. The team used the same process to design and deliver eight products, serving a wide range of functions, in a twelve-month period. Half of those products are generating revenue today, and the rest are awaiting initial orders, all thanks to the power of working in small batches.

THE PROJECT TIME LINE	
Design and engineering of the initial virtual prototype	1 day
Production and assembly of initial hard prototypes	3 days
Design iteration: two additional cycles	5 days
Initial production run and assembly of initial forty units	15 days

## Small Batches in Education

Not every type of product—as it exists today—allows for design

change in small batches. But that is no excuse for sticking to outdated methods. A significant amount of work may be needed to enable innovators to experiment in small batches. As was pointed out in [Chapter 2](#), for established companies looking to accelerate their innovation teams, building this platform for experimentation is the responsibility of senior management.

Imagine that you are a schoolteacher in charge of teaching math to middle school students. Although you may teach concepts in small batches, one day at a time, your overall curriculum cannot change very often. Because you must set up the curriculum in advance and teach the same concepts in the same order to every student in the classroom, you can try a new curriculum at most only once a year.

How could a math teacher experiment with small batches? Under the current large-batch system for educating students, it would be quite difficult; our current educational system was designed in the era of mass production and uses large batches extensively.

A new breed of startups is working hard to change all that. In School of One, students have daily “playlists” of their learning tasks that are attuned to each student’s learning needs, based on that student’s readiness and learning style. For example, Julia is way ahead of grade level in math and learns best in small groups, so her playlist might include three or four videos matched to her aptitude level, a thirty-minute one-on-one tutoring session with her teacher, and a small group activity in which she works on a math puzzle with three peers at similar aptitude levels. There are assessments built into each activity so that data can be fed back to the teacher to choose appropriate tasks for the next playlist. This data can be aggregated across classes, schools, or even whole districts.

Now imagine trying to experiment with a curriculum by using a tool such as School of One. Each student is working at his or her own pace. Let’s say you are a teacher who has a new sequence in mind for how math concepts should be taught. You can see immediately the impact of the change on those of your students who are at that point in the curriculum. If you judge it to be a good change, you could roll it out immediately for every single student:

when they get to that part of the curriculum, they will get the new sequence automatically. In other words, tools like School of One enable teachers to work in much smaller batches, to the benefit of their students. (And, as tools reach wide-scale adoption, successful experiments by individual teachers can be rolled out district-, city-, or even nationwide.) This approach is having an impact and earning accolades. Time magazine recently included School of One in its “most innovative ideas” list; it was the only educational organization to make the list.<sup>5</sup>

## THE LARGE-BATCH DEATH SPIRAL

Small batches pose a challenge to managers steeped in traditional notions of productivity and progress, because they believe that functional specialization is more efficient for expert workers.

Imagine you're a product designer overseeing a new product and you need to produce thirty individual design drawings. It probably seems that the most efficient way to work is in seclusion, by yourself, producing the designs one by one. Then, when you're done with all of them, you pass the drawings on to the engineering team and let them work. In other words, you work in large batches.

From the point of view of individual efficiency, working in large batches makes sense. It also has other benefits: it promotes skill building, makes it easier to hold individual contributors accountable, and, most important, allows experts to work without interruption. At least that's the theory. Unfortunately, reality seldom works out that way.

Consider our hypothetical example. After passing thirty design drawings to engineering, the designer is free to turn his or her attention to the next project. But remember the problems that came up during the envelope-stuffing exercise. What happens when engineering has questions about how the drawings are supposed to work? What if some of the drawings are unclear? What if something goes wrong when engineering attempts to use the drawings?

These problems inevitably turn into interruptions for the designer, and now those interruptions are interfering with the next large batch the designer is supposed to be working on. If the drawings need to be redone, the engineers may become idle while they wait for the rework to be completed. If the designer is not available, the engineers may have to redo the designs themselves. This is why so few products are actually built the way they are designed.

When I work with product managers and designers in companies that use large batches, I often discover that they have to redo their work five or six times for every release. One product manager I worked with was so inundated with interruptions that he took to coming into the office in the middle of the night so that he could work uninterrupted. When I suggested that he try switching the work process from large-batch to single-piece flow, he refused—because that would be inefficient! So strong is the instinct to work in large batches, that even when a large-batch system is malfunctioning, we have a tendency to blame ourselves.

Large batches tend to grow over time. Because moving the batch forward often results in additional work, rework, delays, and interruptions, everyone has an incentive to do work in ever-larger batches, trying to minimize this overhead. This is called the large-batch death spiral because, unlike in manufacturing, there are no physical limits on the maximum size of a batch.<sup>6</sup> It is possible for batch size to keep growing and growing. Eventually, one batch will become the highest-priority project, a “bet the company” new version of the product, because the company has taken such a long time since the last release. But now the managers are incentivized to increase batch size rather than ship the product. In light of how long the product has been in development, why not fix one more bug or add one more feature? Who really wants to be the manager who risked the success of this huge release by failing to address a potentially critical flaw?

I worked at a company that entered this death spiral. We had been working for months on a new version of a really cool product.

The original version had been years in the making, and expectations for the next release were incredibly high. But the longer we worked, the more afraid we became of how customers would react when they finally saw the new version. As our plans became more ambitious, so too did the number of bugs, conflicts, and problems we had to deal with. Pretty soon we got into a situation in which we could not ship anything. Our launch date seemed to recede into the distance. The more work we got done, the more work we had to do. The lack of ability to ship eventually precipitated a crisis and a change of management, all because of the trap of large batches.

These misconceptions about batch size are incredibly common. Hospital pharmacies often deliver big batches of medications to patient floors once a day because it's efficient (a single trip, right?). But many of those meds get sent back to the pharmacy when a patient's orders have changed or the patient is moved or discharged, causing the pharmacy staff to do lots of rework and reprocessing (or trashing) of meds. Delivering smaller batches every four hours reduces the total workload for the pharmacy and ensures that the right meds are at the right place when needed.

Hospital lab blood collections often are done in hourly batches; phlebotomists collect blood for an hour from multiple patients and then send or take all the samples to the lab. This adds to turnaround time for test results and can harm test quality. It has become common for hospitals to bring small batches (two patients) or a single-patient flow of specimens to the lab even if they have to hire an extra phlebotomist or two to do so, because the total system cost is lower.<sup>7</sup>

## PULL, DON'T PUSH

Let's say you are out for a drive, pondering the merits of small batches, and find yourself accidentally putting a dent in your new 2011 blue Toyota Camry. You take it into the dealership for repair and wait to hear the bad news. The repair technician tells you that

you need to have the bumper replaced. He goes to check their inventory levels and tells you he has a new bumper in stock and they can complete your repair immediately. This is good news for everyone—you because you get your car back sooner and the dealership because they have a happy customer and don't run the risk of your taking the car somewhere else for repair. Also, they don't have to store your car or give you a loaner while they wait for the part to come in.

In traditional mass production, the way to avoid stockouts—not having the product the customer wants—is to keep a large inventory of spares just in case. It may be that the blue 2011 Camry bumper is quite popular, but what about last year's model or the model from five years ago? The more inventory you keep, the greater the likelihood you will have the right product in stock for every customer. But large inventories are expensive because they have to be transported, stored, and tracked. What if the 2011 bumper turns out to have a defect? All the spares in all the warehouses instantly become waste.

Lean production solves the problem of stockouts with a technique called pull. When you bring a car into the dealership for repair, one blue 2011 Camry bumper gets used. This creates a “hole” in the dealer's inventory, which automatically causes a signal to be sent to a local restocking facility called the Toyota Parts Distribution Center (PDC). The PDC sends the dealer a new bumper, which creates another hole in inventory. This sends a similar signal to a regional warehouse called the Toyota Parts Redistribution Center (PRC), where all parts suppliers ship their products. That warehouse signals the factory where the bumpers are made to produce one more bumper, which is manufactured and shipped to the PRC.

The ideal goal is to achieve small batches all the way down to single-piece flow along the entire supply chain. Each step in the line pulls the parts it needs from the previous step. This is the famous Toyota just-in-time production method.<sup>8</sup>

When companies switch to this kind of production, their



warehouses immediately shrink, as the amount of just-in-case inventory [called work-in-progress (WIP) inventory] is reduced dramatically. This almost magical shrinkage of WIP is where lean manufacturing gets its name. It's as if the whole supply chain suddenly went on a diet.

Startups struggle to see their work-in-progress inventory. When factories have excess WIP, it literally piles up on the factory floor. Because most startup work is intangible, it's not nearly as visible. For example, all the work that goes into designing the minimum viable product is—until the moment that product is shipped—just WIP inventory. Incomplete designs, not-yet-validated assumptions, and most business plans are WIP. Almost every Lean Startup technique we've discussed so far works its magic in two ways: by converting push methods to pull and reducing batch size. Both have the net effect of reducing WIP.

In manufacturing, pull is used primarily to make sure production processes are tuned to levels of customer demand. Without this, factories can wind up making much more—or much less—of a product than customers really want. However, applying this approach to developing new products is not straightforward. Some people misunderstand the Lean Startup model as simply applying pull to customer wants. This assumes that customers could tell us what products to build and that this would act as the pull signal to product development to make them.<sup>9</sup>

As was mentioned earlier, this is not the way the Lean Startup model works, because customers often don't know what they want. Our goal in building products is to be able to run experiments that will help us learn how to build a sustainable business. Thus, the right way to think about the product development process in a Lean Startup is that it is responding to pull requests in the form of experiments that need to be run.

As soon as we formulate a hypothesis that we want to test, the product development team should be engineered to design and run this experiment as quickly as possible, using the smallest batch size that will get the job done. Remember that although we write the

feedback loop as Build-Measure-Learn because the activities happen in that order, our planning really works in the reverse order: we figure out what we need to learn and then work backwards to see what product will work as an experiment to get that learning. Thus, it is not the customer, but rather our hypothesis about the customer, that pulls work from product development and other functions. Any other work is waste.

## Hypothesis Pull in Clean Tech

To see this in action, let's take a look at Berkeley-based startup Alphabet Energy. Any machine or process that generates power, whether it is a motor in a factory or a coal-burning power plant, generates heat as a by-product. Alphabet Energy has developed a product that can generate electricity from this waste heat, using a new kind of material called a thermoelectric. Alphabet Energy's thermoelectric material was developed over ten years by scientists at the Lawrence Berkeley National Laboratories.

As with many clean technology products, there are huge challenges in bringing a product like this to market. While working through its leap-of-faith assumptions, Alphabet figured out early that developing a solution for waste thermoelectricity required building a heat exchanger and a generic device to transfer heat from one medium to another as well as doing project-specific engineering. For instance, if Alphabet wanted to build a solution for a utility such as Pacific Gas and Electric, the heat exchanger would have to be configured, shaped, and installed to capture the heat from a power plant's exhaust system.

What makes Alphabet Energy unique is that the company made a savvy decision early on in the research process. Instead of using relatively rare elements as materials, they decided to base their research on silicon wafers, the same physical substance that computer central processing units (CPUs) are made from. As CEO Matthew Scullin explains, "Our thermoelectric is the only one that can use low-cost semiconductor infrastructure for manufacturing."

This has enabled Alphabet Energy to design and build its products in small batches.

By contrast, most successful clean technology startups have had to make substantial early investments. The solar panel provider SunPower had to build in factories to manufacture its panels and partner with installers before becoming fully operational. Similarly, BrightSource raised \$291 million to build and operate large-scale solar plants without delivering a watt to a single customer.

Instead of having to invest time and money in expensive fabrication facilities, Alphabet is able to take advantage of the massive existing infrastructure that produces silicon wafers for computer electronics. As a result, Alphabet can go from a product concept to holding a physical version in its hand in just six weeks from end to end. Alphabet's challenge has been to find the combination of performance, price, and physical shape that is a match for early customers. Although its technology has revolutionary potential, early adopters will deploy it only if they can see a clear return on investment.

It might seem that the most obvious market for Alphabet's technology would be power plants, and indeed, that was the team's initial hypothesis. Alphabet hypothesized that simple cycle gas turbines would be an ideal application; these turbines, which are similar to jet engines strapped to the ground, are used by power generators to provide energy for peak demand. Alphabet believed that attaching its semiconductors to those turbines would be simple and cheap.

The company went about testing this hypothesis in small batches by building small-scale solutions for its customers as a way of learning. As with many initial ideas, their hypothesis was disproved quickly. Power companies have a low tolerance for risk, making them unlikely to become early adopters. Because it wasn't weighed down by a large-batch approach, Alphabet was ready to pivot after just three months of investigation.

Alphabet has eliminated many other potential markets as well, leading to a series of customer segment pivots. The company's current efforts are focused on manufacturing firms, which have the

ability to experiment with new technologies in separate parts of their factory; this allows early adopters to evaluate the real-world benefits before committing to a larger deployment. These early deployments are putting more of Alphabet's assumptions to the test. Unlike in the computer hardware business, customers are not willing to pay top dollar for maximum performance. This has required significant changes in Alphabet's product, configuring it to achieve the lowest cost per watt possible.

All this experimentation has cost the company a tiny fraction of what other energy startups have consumed. To date, Alphabet has raised approximately \$1 million. Only time will tell if they will prevail, but thanks to the power of small batches, they will be able to discover the truth much faster.<sup>10</sup>



The Toyota Production System is probably the most advanced system of management in the world, but even more impressive is the fact that Toyota has built the most advanced learning organization in history. It has demonstrated an ability to unleash the creativity of its employees, achieve consistent growth, and produce innovative new products relentlessly over the course of nearly a century.<sup>11</sup>

This is the kind of long-term success to which entrepreneurs should aspire. Although lean production techniques are powerful, they are only a manifestation of a high-functioning organization that is committed to achieving maximum performance by employing the right measures of progress over the long term. Process is only the foundation upon which a great company culture can develop. But without this foundation, efforts to encourage learning, creativity, and innovation will fall flat—as many disillusioned directors of HR can attest.

The Lean Startup works only if we are able to build an organization as adaptable and fast as the challenges it faces. This requires tackling the human challenges inherent in this new way of working; that is the subject of the remainder of [Part Three](#).

## THE STARTUP WAY



# 10

## GROW

I recently had two startups seek my advice on the same day. As types of businesses, they could not have been more different. The first is developing a marketplace to help traders of collectibles connect with one another. These people are hard-core fans of movies, anime, or comics who strive to put together complete collections of toys and other promotional merchandise related to the characters they love. The startup aspires to compete with online marketplaces such as eBay as well as physical marketplaces attached to conventions and other gatherings of fans.

The second startup sells database software to enterprise customers. They have a next-generation database technology that can supplement or replace offerings from large companies such as Oracle, IBM, and SAP. Their customers are chief information officers (CIOs), IT managers, and engineers in some of the world's largest organizations. These are long-lead-time sales that require salespeople, sales engineering, installation support, and maintenance contracts.

You could be forgiven for thinking these two companies have absolutely nothing in common, yet both came to me with the exact same problem. Each one had early customers and promising early revenue. They had validated and invalidated many hypotheses in their business models and were executing against their product road maps successfully. Their customers had provided a healthy mix of positive feedback and suggestions for improvements. Both companies had used their early success to raise money from outside

investors.

The problem was that neither company was growing.

Both CEOs brought me identical-looking graphs showing that their early growth had flatlined. They could not understand why. They were acutely aware of the need to show progress to their employees and investors and came to me because they wanted advice on how to jump-start their growth. Should they invest in more advertising or marketing programs? Should they focus on product quality or new features? Should they try to improve conversion rates or pricing?

As it turns out, both companies share a deep similarity in the way their businesses grow—and therefore a similar confusion about what to do. Both are using the same engine of growth, the topic of this chapter.

## WHERE DOES GROWTH COME FROM?

The engine of growth is the mechanism that startups use to achieve sustainable growth. I use the word sustainable to exclude all one-time activities that generate a surge of customers but have no long-term impact, such as a single advertisement or a publicity stunt that might be used to jump-start growth but could not sustain that growth for the long term.

Sustainable growth is characterized by one simple rule:

New customers come from the actions of past customers.

There are four primary ways past customers drive sustainable growth:

1. Word of mouth. Embedded in most products is a natural level of growth that is caused by satisfied customers' enthusiasm for the product. For example, when I bought my first TiVo DVR, I couldn't stop telling my friends and family about it. Pretty soon, my entire family was using one.

2. As a side effect of product usage. Fashion or status, such as luxury goods products, drive awareness of themselves whenever they are used. When you see someone dressed in the latest clothes or driving a certain car, you may be influenced to buy that product. This is also true of so-called viral products such as Facebook and PayPal. When a customer sends money to a friend using PayPal, the friend is exposed automatically to the PayPal product.

3. Through funded advertising. Most businesses employ advertising to entice new customers to use their products. For this to be a source of sustainable growth, the advertising must be paid for out of revenue, not one-time sources such as investment capital. As long as the cost of acquiring a new customer (the so-called marginal cost) is less than the revenue that customer generates (the marginal revenue), the excess (the marginal profit) can be used to acquire more customers. The more marginal profit, the faster the growth.

4. Through repeat purchase or use. Some products are designed to be purchased repeatedly either through a subscription plan (a cable company) or through voluntary repurchases (groceries or lightbulbs). By contrast, many products and services are intentionally designed as one-time events, such as wedding planning.

These sources of sustainable growth power feedback loops that I have termed engines of growth. Each is like a combustion engine, turning over and over. The faster the loop turns, the faster the company will grow. Each engine has an intrinsic set of metrics that determine how fast a company can grow when using it.

## THE THREE ENGINES OF GROWTH



We saw in [Part Two](#) how important it is for startups to use the right kind of metrics—actionable metrics—to evaluate their progress. However, this leaves a large amount of variety in terms of which numbers one should measure. In fact, one of the most expensive forms of potential waste for a startup is spending time arguing about how to prioritize new development once it has a product on the market. At any time, the company could invest its energy in finding new customers, servicing existing customers better, improving overall quality, or driving down costs. In my experience, the discussions about these kinds of priority decisions can consume a substantial fraction of the company's time.

Engines of growth are designed to give startups a relatively small set of metrics on which to focus their energies. As one of my mentors, the venture capital investor Shawn Carolan, put it, “Startups don't starve; they drown.” There are always a zillion new ideas about how to make the product better floating around, but the hard truth is that most of those ideas make a difference only at the margins. They are mere optimizations. Startups have to focus on the big experiments that lead to validated learning. The engines of growth framework helps them stay focused on the metrics that matter.

## The Sticky Engine of Growth

This brings us back to the two startups that kicked off this chapter. Both are using the exact same engine of growth despite being in very different industries. Both products are designed to attract and retain customers for the long term. The underlying mechanism of that retention is different in the two cases. For the collectible company, the idea is to become the number one shopping destination for fanatical collectors. These are people who are constantly hunting for the latest items and the best deals. If the company's product works as designed, collectors who start using it will check constantly and repeatedly to see if new items are for sale as well as listing their own items for sale or trade.

The startup database vendor relies on repeat usage for a very different reason. Database technology is used only as the foundation for a customer's own products, such as a website or a point of sale system. Once you build a product on top of a particular database technology, it is extremely difficult to switch. In the IT industry, such customers are said to be locked in to the vendor they choose. For such a product to grow, it has to offer such a compelling new capability that customers are willing to risk being tied to a proprietary vendor for a potentially long time.

Thus, both businesses rely on having a high customer retention rate. They have an expectation that once you start using their product, you will continue to do so. This is the same dynamic as a mobile telephone service provider: when a customer cancels his or her service, it generally means that he or she is extremely dissatisfied or is switching to a competitor's product. This is in contrast to, say, groceries on a store aisle. In the grocery retail business, customer tastes fluctuate, and if a customer buys a Pepsi this week instead of Coke, it's not necessarily a big deal.

Therefore, companies using the sticky engine of growth track their attrition rate or churn rate very carefully. The churn rate is defined as the fraction of customers in any period who fail to remain engaged with the company's product.

The rules that govern the sticky engine of growth are pretty simple: if the rate of new customer acquisition exceeds the churn rate, the product will grow. The speed of growth is determined by what I call the rate of compounding, which is simply the natural growth rate minus the churn rate. Like a bank account that earns compounding interest, having a high rate of compounding will lead to extremely rapid growth—without advertising, viral growth, or publicity stunts.

Unfortunately, both of these sticky startups were tracking their progress using generic indicators such as the total number of customers. Even the actionable metrics they were using, such as the activation rate and revenue per customer, weren't very helpful because in the sticky engine of growth, these variables have little impact on growth. (In the sticky engine of growth, they are better

sued to testing the value hypothesis that was discussed in [Chapter 5](#).)

After our meeting, one of the two startups took me up on my advice to model its customer behavior by using the sticky engine of growth as a template. The results were striking: a 61 percent retention rate and a 39 percent growth rate of new customers. In other words, its churn rate and new customer acquisition balanced each other almost perfectly, leading to a compounding growth rate of just 0.02 percent—almost zero.

This is typical for companies in an engagement business that are struggling to find growth. An insider who worked at the dot-com-era company PointCast once showed me how that company suffered a similar dysfunction. When PointCast was struggling to grow, it was nonetheless incredibly successful in new customer acquisition—just like this sticky startup (39 percent every period). Unfortunately, this growth is being offset by an equivalent amount of churn. Once it is modeled this way, the good news should be apparent: there are plenty of new customers coming in the door. The way to find growth is to focus on existing customers for the product even more engaging to them. For example, the company could focus on getting more and better listings. This would create an incentive for customers to check back often. Alternatively, the company could do something more direct such as messaging them about limited-time sales or special offers. Either way, its focus needs to be on improving customer retention. This goes against the standard intuition in that if a company lacks growth, it should invest more in sales and marketing. This counterintuitive result is hard to infer from standard vanity metrics.

## The Viral Engine of Growth

Online social networks and Tupperware are examples of products for which customers do the lion's share of the marketing. Awareness of the product spreads rapidly from person to person similarly to the way a virus becomes an epidemic. This is distinct from the

simple word-of-mouth growth discussed above. Instead, products that exhibit viral growth depend on person-to-person transmission as a necessary consequence of normal product use. Customers are not intentionally acting as evangelists; they are not necessarily trying to spread the word about the product. Growth happens automatically as a side effect of customers using the product. Viruses are not optional.

For example, one of the most famous viral success stories is a company called Hotmail. In 1996, Sabeer Bhatia and Jack Smith launched a new web-based e-mail service that offered customers free accounts. At first, growth was sluggish; with only a small seed investment from the venture capital firm Draper Fisher Jurvetson, the Hotmail team could not afford an extensive marketing campaign. But everything changed when they made one small tweak to the product. They added to the bottom of every single e-mail the message “P.S. Get your free e-mail at Hotmail” along with a clickable link.

Within weeks, that small product change produced massive results. Within six months, Bhatia and Smith had signed up more than 1 million new customers. Five weeks later, they hit the 2 million mark. Eighteen months after launching the service, with 12 million subscribers, they sold the company to Microsoft for \$400 million.<sup>1</sup>

The same phenomenon is at work in Tupperware’s famous “house parties,” in which customers earn commissions by selling the product to their friends and neighbors. Every sales pitch is an opportunity not only to sell Tupperware products but also to persuade other customers to become Tupperware representatives. Tupperware parties are still going strong decades after they started. Many other contemporary companies, such as Pampered Chef (owned by Warren Buffett’s Berkshire Hathaway), Southern Living, and Tastefully Simple, have adopted a similar model successfully.

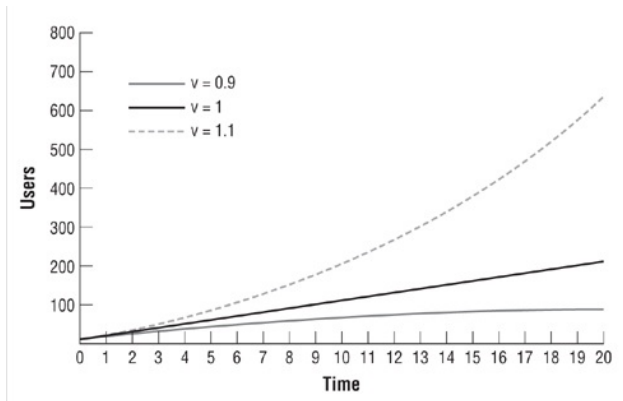
Like the other engines of growth, the viral engine is powered by a feedback loop that can be quantified. It is called the viral loop, and its speed is determined by a single mathematical term called

the viral coefficient. The higher this coefficient is, the faster the product will spread. The viral coefficient measures how many new customers will use a product as a consequence of each new customer who signs up. Put another way, how many friends will each customer bring with him or her? Since each friend is also a new customer, he or she has an opportunity to recruit yet more friends.

For a product with a viral coefficient of 0.1, one in every ten customers will recruit one of his or her friends. This is not a sustainable loop. Imagine that one hundred customers sign up. They will cause ten friends to sign up. Those ten friends will cause one additional person to sign up, but there the loop will fizzle out.

By contrast, a viral loop with a coefficient that is greater than 1.0 will grow exponentially, because each person who signs up will bring, on average, more than one other person with him or her.

To see these effects graphically, take a look at this chart:



Companies that rely on the viral engine of growth must focus on increasing the viral coefficient more than anything else, because

even tiny changes in this number will cause dramatic changes in their future prospects.

A consequence of this is that many viral products do not charge customers directly but rely on indirect sources of revenue such as advertising. This is the case because viral products cannot afford to have any friction impede the process of signing customers up and recruiting their friends. This can make testing the value hypothesis for viral products especially challenging.

The true test of the value hypothesis is always a voluntary exchange of value between customers and the startup that serves them. A lot of confusion stems from the fact that this exchange can be monetary, as in the case of Tupperware, or nonmonetary, as in the case of Facebook. In the viral engine of growth, monetary exchange does not drive new growth; it is useful only as an indicator that customers value the product enough to pay for it. If Facebook or Hotmail had started charging customers in their early days, it would have been foolish, as it would have impeded their ability to grow. However, it is not true that customers do not give these companies something of value: by investing their time and attention in the product, they make the product valuable to advertisers. Companies that sell advertising actually serve two different groups of customers—consumers and advertisers—and exchange a different currency of value with each.<sup>2</sup>

This is markedly different from companies that actively use money to fuel their expansion, such as a retail chain that can grow as fast as it can fund the opening of new stores at suitable locations. These companies are using a different engine of growth altogether.

## The Paid Engine of Growth

Imagine another pair of businesses. The first makes \$1 on each customer it signs up; the second makes \$100,000 from each customer it signs up. To predict which company will grow faster, you need to know only one additional thing: how much it costs to sign up a new customer.

Imagine that the first company uses Google AdWords to find new customers online and pays an average of 80 cents each time a new customer joins. The second company sells heavy goods to large companies. Each sale requires a significant time investment from a salesperson and on-site sales engineering to help install the product; these hard costs total up to \$80,000 per new customer. Both companies will grow at the exact same rate. Each has the same proportion of revenue (20 percent) available to reinvest in new customer acquisition. If either company wants to increase its rate of growth, it can do so in one of two ways: increase the revenue from each customer or drive down the cost of acquiring a new customer.

That's the paid engine of growth at work.

In relating the IMVU story in [Chapter 3](#), I talked about how we made a major early mistake in setting up the IMVU strategy. We ultimately wound up having to make an engine of growth pivot. We originally thought that our IM add-on strategy would allow the product to grow virally. Unfortunately, customers refused to go along with our brilliant strategy.

Our basic misconception was a belief that customers would be willing to use IMVU as an add-on to existing instant messaging networks. We believed that the product would spread virally through those networks, passed from customer to customer. The problem with that theory is that some kinds of products are not compatible with viral growth.

IMVU's customers didn't want to use the product with their existing friends. They wanted to use it to make new friends. Unfortunately, that meant they did not have a strong incentive to bring new customers to the product; they viewed that as our job. Fortunately, IMVU was able to grow by using paid advertising because our customers were willing to pay more for our product than it cost us to reach them via advertising.

Like the other engines, the paid engine of growth is powered by a feedback loop. Each customer pays a certain amount of money for the product over his or her "lifetime" as a customer. Once variable costs are deducted, this usually is called the customer lifetime value (LTV). This revenue can be invested in growth by buying

advertising.

Suppose an advertisement costs \$100 and causes fifty new customers to sign up for the service. This ad has a cost per acquisition (CPA) of \$2.00. In this example, if the product has an LTV that is greater than \$2, the product will grow. The margin between the LTV and the CPA determines how fast the paid engine of growth will turn (this is called the marginal profit). Conversely, if the CPA remains at \$2.00 but the LTV falls below \$2.00, the company's growth will slow. It may make up the difference with one-time tactics such as using invested capital or publicity stunts, but those tactics are not sustainable. This was the fate of many failed companies, including notable dot-com flameouts that erroneously believed that they could lose money on each customer but, as the old joke goes, make it up in volume.

Although I have explained the paid engine of growth in terms of advertising, it is far broader than that. Startups that employ an outbound sales force are also using this engine, as are retail companies that rely on foot traffic. All these costs should be factored into the cost per acquisition.

For example, one startup I worked with built collaboration tools for teams and groups. It went through a radical pivot, switching from a tool that was used primarily by hobbyists and small clubs to one that was sold primarily to enterprises, nongovernmental organizations (NGOs), and other extremely large organizations. However, they made that customer segment pivot without changing their engine of growth. Previously, they had done customer acquisition online, using web-based direct marketing techniques. I remember one early situation in which the company fielded a call from a major NGO that wanted to buy its product and roll it out across many divisions. The startup had an "unlimited" pricing plan, its most expensive, that cost only a few hundred dollars per month. The NGO literally could not make the purchase because it had no process in place for buying something so inexpensive. Additionally, the NGO needed substantial help in managing the rollout, educating its staff on the new tool, and tracking the impact of the change; those were all services the company was ill equipped to offer.



Changing customer segments required them to switch to hiring a sizable outbound sales staff that spent time attending conferences, educating executives, and authoring white papers. Those much higher costs came with a corresponding reward: the company switched from making only a few dollars per customer to making tens and then hundreds of thousands of dollars per much larger customer. Their new engine of growth led to sustained success.

Most sources of customer acquisition are subject to competition. For example, prime retail storefronts have more foot traffic and are therefore more valuable. Similarly, advertising that is targeted to more affluent customers generally costs more than advertising that reaches the general public. What determines these prices is the average value earned in aggregate by the companies that are in competition for any given customer's attention. Wealthy consumers cost more to reach because they tend to become more profitable customers.

Over time, any source of customer acquisition will tend to have its CPA bid up by this competition. If everyone in an industry makes the same amount of money on each sale, they all will wind up paying most of their marginal profit to the source of acquisition. Thus, the ability to grow in the long term by using the paid engine requires a differentiated ability to monetize a certain set of customers.

IMVU is a case in point. Our customers were not considered very lucrative by other online services: they included a lot of teenagers, low-income adults, and international customers. Other services tended to assume those people would not pay for anything online. At IMVU, we developed techniques for collecting online payments from customers who did not have a credit card, such as allowing them to bill to their mobile phones or send us cash in the mail. Therefore, we could afford to pay more to acquire those customers than our competitors could.

## A Technical Caveat

Technically, more than one engine of growth can operate in a business at a time. For example, there are products that have extremely fast viral growth as well as extremely low customer churn rates. Also, there is no reason why a product cannot have both high margins and high retention. However, in my experience, successful startups usually focus on just one engine of growth, specializing in everything that is required to make it work. Companies that attempt to build a dashboard that includes all three engines tend to cause a lot of confusion because the operations expertise required to model all these effects simultaneously is quite complicated. Therefore, I strongly recommend that startups focus on one engine at a time. Most entrepreneurs already have a strong leap-of-faith hypothesis about which engine is most likely to work. If they do not, time spent out of the building with customers will quickly suggest one that seems profitable. Only after pursuing one engine thoroughly should a startup consider a pivot to one of the others.

## ENGINES OF GROWTH DETERMINE PRODUCT/MARKET FIT

Marc Andreessen, the legendary entrepreneur and investor and one of the fathers of the World Wide Web, coined the term product/market fit to describe the moment when a startup finally finds a widespread set of customers that resonate with its product:

In a great market—a market with lots of real potential customers—the market pulls product out of the startup. This is the story of search keyword advertising, Internet auctions, and TCP/IP routers. Conversely, in a terrible market, you can have the best product in the world and an absolutely killer team, and it doesn't matter—you're going to fail.<sup>3</sup>

When you see a startup that has found a fit with a large market, it's exhilarating. It leaves no room for doubt. It is Ford's Model T

flying out of the factory as fast as it could be made, Facebook sweeping college campuses practically overnight, or Lotus taking the business world by storm, selling \$54 million worth of Lotus 1-2-3 in its first year of operation.

Startups occasionally ask me to help them evaluate whether they have achieved product/market fit. It's easy to answer: if you are asking, you're not there yet. Unfortunately, this doesn't help companies figure out how to get closer to product/market fit. How can you tell if you are on the verge of success or hopelessly far away?

Although I don't think Andreessen intended this as part of his definition, to many entrepreneurs it implies that a pivot is a failure event—"our startup has failed to achieve product/market fit." It also implies the inverse—that once our product has achieved product/market fit, we won't have to pivot anymore. Both assumptions are wrong.

I believe the concept of the engine of growth can put the idea of product/market fit on a more rigorous footing. Since each engine of growth can be defined quantitatively, each has a unique set of metrics that can be used to evaluate whether a startup is on the verge of achieving product/market fit. A startup with a viral coefficient of 0.9 or more is on the verge of success. Even better, the metrics for each engine of growth work in tandem with the innovation accounting model discussed in [Chapter 7](#) to give direction to a startup's product development efforts. For example, if a startup is attempting to use the viral engine of growth, it can focus its development efforts on things that might affect customer behavior—on the viral loop—and safely ignore those that do not. Such a startup does not need to specialize in marketing, advertising, or sales functions. Conversely, a company using the paid engine needs to develop those marketing and sales functions urgently.

A startup can evaluate whether it is getting closer to product/market fit as it tunes its engine by evaluating each trip through the Build-Measure-Learn feedback loop using innovation accounting. What really matters is not the raw numbers or vanity metrics but the direction and degree of progress.

For example, imagine two startups that are working diligently to tune the sticky engine of growth. One has a compounding rate of growth of 5 percent, and the other 10 percent. Which company is the better bet? On the surface, it may seem that the larger rate of growth is better, but what if each company's innovation accounting dashboard looks like the following chart?

COMPOUNDING GROWTH RATE AS OF	COMPANY A	COMPANY B
Six months ago	0.1%	9.8%
Five months ago	0.5%	9.6%
Four months ago	2.0%	9.9%
Three months ago	3.2%	9.8%
Two months ago	4.5%	9.7%
One month ago	5.0%	10.0%

Even with no insight into these two companies' gross numbers, we can tell that company A is making real progress whereas company B is stuck in the mud. This is true even though company B is growing faster than company A right now.

## WHEN ENGINES RUN OUT

Getting a startup's engine of growth up and running is hard enough, but the truth is that every engine of growth eventually runs out of gas. Every engine is tied to a given set of customers and their related habits, preferences, advertising channels, and interconnections. At some point, that set of customers will be exhausted. This may take a long time or a short time, depending on one's industry and timing.

**Chapter 6** emphasized the importance of building the minimum viable product in such a way that it contains no additional features beyond what is required by early adopters. Following that strategy successfully will unlock an engine of growth that can reach that target audience. However, making the transition to mainstream customers will require tremendous additional work.<sup>4</sup> Once we have a product that is growing among early adopters, we could in theory stop work in product development entirely. The product would continue to grow until it reached the limits of that early market. Then growth would level off or even stop completely. The challenge comes from the fact that this slowdown might take months or even years to take place. Recall from **Chapter 8** that IMVU failed this test—at first—for precisely this reason.

Some unfortunate companies wind up following this strategy inadvertently. Because they are using vanity metrics and traditional accounting, they think they are making progress when they see their numbers growing. They falsely believe they are making their product better when in fact they are having no impact on customer behavior. The growth is all coming from an engine of growth that is working—running efficiently to bring in new customers—not from improvements driven by product development. Thus, when the growth suddenly slows, it provokes a crisis.

This is the same problem that established companies experience. Their past successes were built on a finely tuned engine of growth. If that engine runs its course and growth slows or stops, there can be a crisis if the company does not have new startups incubating within its ranks that can provide new sources of growth.

Companies of any size can suffer from this perpetual affliction. They need to manage a portfolio of activities, simultaneously tuning their engine of growth and developing new sources of growth for when that engine inevitably runs its course. How to do this is the subject of **Chapter 12**. However, before we can manage that portfolio, we need an organizational structure, culture, and discipline that can handle these rapid and often unexpected changes. I call this an adaptive organization, and it is the subject of



# 11

## ADAPT

When I was the CTO of IMVU, I thought I was doing a good job most of the time. I had built an agile engineering organization, and we were successfully experimenting with the techniques that would come to be known as the Lean Startup. However, on a couple of occasions I suddenly realized that I was failing at my job. For an achievement-oriented person, that is incredibly disarming. Worst of all, you don't get a memo. If you did, it would read something like this:

Dear Eric,

Congratulations! The job you used to do at this company is no longer available. However, you have been transferred to a new job in the company. Actually, it's not the same company anymore, even though it has the same name and many of the same people. And although the job has the same title, too, and you used to be good at your old job, you're already failing at the new one. This transfer is effective as of six months ago, so this is to alert you that you've already been failing at it for quite some time.

Best of luck!

Every time this happened to me, I struggled to figure out what to do. I knew that as the company grew, we would need additional processes and systems designed to coordinate the company's operations at each larger size. And yet I had also seen many startups

become ossified and bureaucratic out of a misplaced desire to become “professional.”

Having no system at all was not an option for IMVU and is not an option for you. There are so many ways for a startup to fail. I’ve lived through the overarchitecture failure, in which attempting to prevent all the various kinds of problems that could occur wound up delaying the company from putting out any product. I’ve seen companies fail the other way from the so-called Friendster effect, suffering a high-profile technical failure just when customer adoption is going wild. As a department executive, this outcome is worst of all, because the failure is both high-profile and attributable to a single function or department—yours. Not only will the company fail, it will be your fault.

Most of the advice I’ve heard on this topic has suggested a kind of split-the-difference approach (as in, “engage in a little planning, but not too much”). The problem with this willy-nilly approach is that it’s hard to give any rationale for why we should anticipate one particular problem but ignore another. It can feel like the boss is being capricious or arbitrary, and that feeds the common feeling that management’s decisions conceal an ulterior motive.

For those being managed this way, their incentives are clear. If the boss tends to split the difference, the best way to influence the boss and get what you want is to take the most extreme position possible. For example, if one group is advocating for an extremely lengthy release cycle, say, an annual new product introduction, you might choose to argue for an equally extremely short release cycle (perhaps weekly or even daily), knowing that the two opinions will be averaged out. Then, when the difference is split, you’re likely to get an outcome closer to what you actually wanted in the first place. Unfortunately, this kind of arms race escalates. Rivals in another camp are likely to do the same thing. Over time, everyone will take the most polarized positions possible, which makes splitting the difference ever more difficult and ever less successful. Managers have to take responsibility for knowingly or inadvertently creating such incentives. Although it was not their intention to reward extreme polarization, that’s exactly what they are doing.



Getting out of this trap requires a significant shift in thinking.

## BUILDING AN ADAPTIVE ORGANIZATION

Should a startup invest in a training program for new employees? If you had asked me a few years ago, I would have laughed and said, “Absolutely not. Training programs are for big companies that can afford them.” Yet at IMVU we wound up building a training program that was so good, new hires were productive on their first day of employment. Within just a few weeks, those employees were contributing at a high level. It required a huge effort to standardize our work processes and prepare a curriculum of the concepts that new employees should learn. Every new engineer would be assigned a mentor, who would help the new employee work through a curriculum of systems, concepts, and techniques he or she would need to become productive at IMVU. The performance of the mentor and mentee were linked, so the mentors took this education seriously.

What is interesting, looking back at this example, is that we never stopped work and decided that we needed to build a great training program. Instead, the training program evolved organically out of a methodical approach to evolving our own process. This process of orientation was subject to constant experimentation and revision so that it grew more effective—and less burdensome—over time.

I call this building an adaptive organization, one that automatically adjusts its process and performance to current conditions.

### Can You Go Too Fast?

So far this book has emphasized the importance of speed. Startups are in a life-or-death struggle to learn how to build a sustainable business before they run out of resources and die. However, focusing on speed alone would be destructive. To work, startups

require built-in speed regulators that help teams find their optimal pace of work.

We saw an example of speed regulation in [Chapter 9](#) with the use of the andon cord in systems such as continuous deployment. It is epitomized in the paradoxical Toyota proverb, “Stop production so that production never has to stop.” The key to the andon cord is that it brings work to a stop as soon as an uncorrectable quality problem surfaces—which forces it to be investigated. This is one of the most important discoveries of the lean manufacturing movement: you cannot trade quality for time. If you are causing (or missing) quality problems now, the resulting defects will slow you down later. Defects cause a lot of rework, low morale, and customer complaints, all of which slow progress and eat away at valuable resources.

So far I have used the language of physical products to describe these problems, but that is simply a matter of convenience. Service businesses have the same challenges. Just ask any manager of a training, staffing, or hospitality firm to show you the playbook that specifies how employees are supposed to deliver the service under various conditions. What might have started out as a simple guide tends to grow inexorably over time. Pretty soon, orientation is incredibly complex and employees have invested a lot of time and energy in learning the rules. Now consider an entrepreneurial manager in that kind of company trying to experiment with new rules or procedures. The higher-quality the existing playbook is, the easier it will be for it to evolve over time. By contrast, a low-quality playbook will be filled with contradictory or ambiguous rules that cause confusion when anything is changed.

When I teach the Lean Startup approach to entrepreneurs with an engineering background, this is one of the hardest concepts to grasp. On the one hand, the logic of validated learning and the minimum viable product says that we should get a product into customers’ hands as soon as possible and that any extra work we do beyond what is required to learn from customers is waste. On the other hand, the Build-Measure-Learn feedback loop is a continuous process. We don’t stop after one minimum viable product but use

what we have learned to get to work immediately on the next iteration.

Therefore, shortcuts taken in product quality, design, or infrastructure today may wind up slowing a company down tomorrow. You can see this paradox in action at IMVU. [Chapter 3](#) recounted how we wound up shipping a product to customers that was full of bugs, missing features, and bad design. The customers wouldn't even try that product, and so most of that work had to be thrown away. It's a good thing we didn't waste a lot of time fixing those bugs and cleaning up that early version.

However, as our learning allowed us to build products that customers did want, we faced slowdowns. Having a low-quality product can inhibit learning when the defects prevent customers from experiencing (and giving feedback on) the product's benefits. In IMVU's case, as we offered the product to more mainstream customers, they were much less forgiving than early adopters had been. Similarly, the more features we added to the product, the harder it became to add even more because of the risk that a new feature would interfere with an existing feature. The same dynamics happen in a service business, since any new rules may conflict with existing rules, and the more rules, the more possibilities for conflict.

IMVU used the techniques of this chapter to achieve scale and quality in a just-in-time fashion.

## THE WISDOM OF THE FIVE WHYS

To accelerate, Lean Startups need a process that provides a natural feedback loop. When you're going too fast, you cause more problems. Adaptive processes force you to slow down and invest in preventing the kinds of problems that are currently wasting time. As those preventive efforts pay off, you naturally speed up again.

Let's return to the question of having a training program for new employees. Without a program, new employees will make mistakes while in their learning curve that will require assistance and intervention from other team members, slowing everyone down.

How do you decide if the investment in training is worth the benefit of speed due to reduced interruptions? Figuring this out from a top-down perspective is challenging, because it requires estimating two completely unknown quantities: how much it will cost to build an unknown program against an unknown benefit you might reap. Even worse, the traditional way to make these kinds of decisions is decidedly large-batch thinking. A company either has an elaborate training program or it does not. Until they can justify the return on investment from building a full program, most companies generally do nothing.

The alternative is to use a system called the Five Whys to make incremental investments and evolve a startup's processes gradually. The core idea of Five Whys is to tie investments directly to the prevention of the most problematic symptoms. The system takes its name from the investigative method of asking the question "Why?" five times to understand what has happened (the root cause). If you've ever had to answer a precocious child who wants to know "Why is the sky blue?" and keeps asking "Why?" after each answer, you're familiar with it. This technique was developed as a systematic problem-solving tool by Taiichi Ohno, the father of the Toyota Production System. I have adapted it for use in the Lean Startup model with a few changes designed specifically for startups.

At the root of every seemingly technical problem is a human problem. Five Whys provides an opportunity to discover what that human problem might be. Taiichi Ohno gives the following example:

When confronted with a problem, have you ever stopped and asked why five times? It is difficult to do even though it sounds easy. For example, suppose a machine stopped functioning:

1. Why did the machine stop? (There was an overload and the fuse blew.)
2. Why was there an overload? (The bearing was not sufficiently lubricated.)

3. Why was it not lubricated sufficiently? (The lubrication pump was not pumping sufficiently.)
4. Why was it not pumping sufficiently? (The shaft of the pump was worn and rattling.)
5. Why was the shaft worn out? (There was no strainer attached and metal scrap got in.)

Repeating “why” five times, like this, can help uncover the root problem and correct it. If this procedure were not carried through, one might simply replace the fuse or the pump shaft. In that case, the problem would recur within a few months. The Toyota production system has been built on the practice and evolution of this scientific approach. By asking and answering “why” five times, we can get to the real cause of the problem, which is often hidden behind more obvious symptoms.<sup>1</sup>

Note that even in Ohno’s relatively simple example the root cause moves away from a technical fault (a blown fuse) and toward a human error (someone forgot to attach a strainer). This is completely typical of most problems that startups face no matter what industry they are in. Going back to our service business example, most problems that at first appear to be individual mistakes can be traced back to problems in training or the original playbook for how the service is to be delivered.

Let me demonstrate how using the Five Whys allowed us to build the employee training system that was mentioned earlier. Imagine that at IMVU we suddenly start receiving complaints from customers about a new version of the product that we have just released.

1. A new release disabled a feature for customers. Why? Because a particular server failed.
2. Why did the server fail? Because an obscure subsystem was used in the wrong way.

3. Why was it used in the wrong way? The engineer who used it didn't know how to use it properly.
4. Why didn't he know? Because he was never trained.
5. Why wasn't he trained? Because his manager doesn't believe in training new engineers because he and his team are "too busy."

What began as a purely technical fault is revealed quickly to be a very human managerial issue.

## Make a Proportional Investment

Here's how to use Five Whys analysis to build an adaptive organization: consistently make a proportional investment at each of the five levels of the hierarchy. In other words, the investment should be smaller when the symptom is minor and larger when the symptom is more painful. We don't make large investments in prevention unless we're coping with large problems.

In the example above, the answer is to fix the server, change the subsystem to make it less error-prone, educate the engineer, and, yes, have a conversation with the engineer's manager.

This latter piece, the conversation with the manager, is always hard, especially in a startup. When I was a startup manager, if you told me I needed to invest in training my people, I would have told you it was a waste of time. There were always too many other things to do. I'd probably have said something sarcastic like "Sure, I'd be happy to do that—if you can spare my time for the eight weeks it'll take to set up." That's manager-speak for "No way in hell."

That's why the proportional investment approach is so important. If the outage is a minor glitch, it's essential that we make only a minor investment in fixing it. Let's do the first hour of the eight-week plan. That may not sound like much, but it's a start. If the problem recurs, asking the Five Whys will require that we continue to make progress on it. If the problem does not occur again, an hour isn't a big loss.

I used the example of engineering training because that was something I was reluctant to invest in at IMVU. At the outset of our venture, I thought we needed to focus all of our energies on building and marketing our product. Yet once we entered a period of rapid hiring, repeated Five Whys sessions revealed that problems caused by lack of training were slowing down product development. At no point did we drop everything to focus solely on training. Instead, we made incremental improvements to the process constantly, each time reaping incremental benefits. Over time, those changes compounded, freeing up time and energy that previously had been lost to firefighting and crisis management.

## Automatic Speed Regulator

The Five Whys approach acts as a natural speed regulator. The more problems you have, the more you invest in solutions to those problems. As the investments in infrastructure or process pay off, the severity and number of crises are reduced and the team speeds up again. With startups in particular, there is a danger that teams will work too fast, trading quality for time in a way that causes sloppy mistakes. Five Whys prevents that, allowing teams to find their optimal pace.

The Five Whys ties the rate of progress to learning, not just execution. Startup teams should go through the Five Whys whenever they encounter any kind of failure, including technical faults, failures to achieve business results, or unexpected changes in customer behavior.

Five Whys is a powerful organizational technique. Some of the engineers I have trained to use it believe that you can derive all the other Lean Startup techniques from the Five Whys. Coupled with working in small batches, it provides the foundation a company needs to respond quickly to problems as they appear, without overinvesting or overengineering.

## THE CURSE OF THE FIVE BLAMES

When teams first adopt Five Whys as a problem-solving tool, they encounter some common pitfalls. We need systems like Five Whys to overcome our psychological limitations because we tend to overreact to what's happening in the moment. We also tend to get frustrated if things happen that we did not anticipate.

When the Five Whys approach goes awry, I call it the Five Blames. Instead of asking why repeatedly in an attempt to understand what went wrong, frustrated teammates start pointing fingers at each other, trying to decide who is at fault. Instead of using the Five Whys to find and fix problems, managers and employees can fall into the trap of using the Five Blames as a means for venting their frustrations and calling out colleagues for systemic failures. Although it's human nature to assume that when we see a mistake, it's due to defects in someone else's department, knowledge, or character, the goal of the Five Whys is to help us see the objective truth that chronic problems are caused by bad process, not bad people, and remedy them accordingly.

I recommend several tactics for escaping the Five Blames. The first is to make sure that everyone affected by the problem is in the room during the analysis of the root cause. The meeting should include anyone who discovered or diagnosed the problem, including customer service representatives who fielded the calls, if possible. It should include anyone who tried to fix the symptom as well as anyone who worked on the subsystems or features involved. If the problem was escalated to senior management, the decision makers who were involved in the escalation should be present as well.

This may make for a crowded room, but it's essential. In my experience, whoever is left out of the discussion ends up being the target for blame. This is just as damaging whether the scapegoat is a junior employee or the CEO. When it's a junior employee, it's all too easy to believe that that person is replaceable. If the CEO is not present, it's all too easy to assume that his or her behavior is unchangeable. Neither presumption is usually correct.



When blame inevitably arises, the most senior people in the room should repeat this mantra: if a mistake happens, shame on us for making it so easy to make that mistake. In a Five Whys analysis, we want to have a systems-level view as much as possible.

Here's a situation in which this mantra came in handy. Because of the training process we had developed at IMVU through the Five Whys, we routinely asked new engineers to make a change to the production environment on their first day. For engineers trained in traditional development methods, this was often frightening. They would ask, "What will happen to me if I accidentally disrupt or stop the production process?" In their previous jobs, that was a mistake that could get them fired. At IMVU we told new hires, "If our production process is so fragile that you can break it on your very first day of work, shame on us for making it so easy to do so." If they did manage to break it, we immediately would have them lead the effort to fix the problem as well as the effort to prevent the next person from repeating their mistake.

For new hires who came from companies with a very different culture, this was often a stressful initiation, but everyone came through it with a visceral understanding of our values. Bit by bit, system by system, those small investments added up to a robust product development process that allowed all our employees to work more creatively, with greatly reduced fear.

## Getting Started

Here are a few tips on how to get started with the Five Whys that are based on my experience introducing this technique at many other companies.

For the Five Whys to work properly, there are rules that must be followed. For example, the Five Whys requires an environment of mutual trust and empowerment. In situations in which this is lacking, the complexity of Five Whys can be overwhelming. In such situations, I've often used a simplified version that still allows teams to focus on analyzing root causes while developing the muscles

they'll need later to tackle the full-blown method.

I ask teams to adopt these simple rules:

1. Be tolerant of all mistakes the first time.
2. Never allow the same mistake to be made twice.

The first rule encourages people to get used to being compassionate about mistakes, especially the mistakes of others. Remember, most mistakes are caused by flawed systems, not bad people. The second rule gets the team started making proportional investments in prevention.

This simplified system works well. In fact, we used it at IMVU in the days before I discovered the Five Whys and the Toyota Production System. However, such a simplified system does not work effectively over the long term, as I found out firsthand. In fact, that was one of the things that drove me to first learn about lean production.

The strength and weakness of the simplified system is that it invites questions such as What counts as the same problem? What kinds of mistakes should we focus on? and Should we fix this individual problem or try to prevent a whole category of related problems? For a team that is just getting started, these questions are thought-provoking and can lay the groundwork for more elaborate methods to come. Ultimately, though, they do need answering. They need a complete adaptive process such as the Five Whys.

## Facing Unpleasant Truths

You will need to be prepared for the fact that Five Whys is going to turn up unpleasant facts about your organization, especially at the beginning. It is going to call for investments in prevention that come at the expense of time and money that could be invested in new products or features. Under pressure, teams may feel that they don't have time to waste on analyzing root causes even though it would give them more time in the long term. The process

sometimes will devolve into the Five Blames. At all these junctures, it is essential that someone with sufficient authority be present to insist that the process be followed, that its recommendations be implemented, and to act as a referee if disagreements flare up. Building an adaptive organization, in other words, requires executive leadership to sponsor and support the process.

Often, individual contributors at startups come to my workshops, eager to get started with the Five Whys. I caution against attempting to do that if they do not have the buy-in of the manager or team leader. Proceed cautiously if you find yourself in this situation. It may not be possible to get the entire team together for a true Five Whys inquiry, but you can always follow the simple two-rule version in your own work. Whenever something goes wrong, ask yourself: How could I prevent myself from being in this situation ever again?

## Start Small, Be Specific

Once you are ready to begin, I recommend starting with a narrowly targeted class of symptoms. For example, the first time I used the Five Whys successfully, I used it to diagnose problems with one of our internal testing tools that did not affect customers directly. It may be tempting to start with something large and important because that is where most of the time is being wasted as a result of a flawed process, but it is also where the pressure will be greatest. When the stakes are high, the Five Whys can devolve into the Five Blames quickly. It's better to give the team a chance to learn how to do the process first and then expand into higher-stakes areas later.

The more specific the symptoms are, the easier it will be for everyone to recognize when it's time to schedule a Five Whys meeting. Say you want to use the Five Whys to address billing complaints from customers. In that case, pick a date after which all billing complaints will trigger a Five Whys meeting automatically. Note that this requires that there be a small enough volume of complaints that having this meeting every time one comes in is

practical. If there are already too many complaints, pick a subset on which you want to focus. Make sure that the rule that determines which kinds of complaints trigger a Five Whys meeting is simple and ironclad. For example, you might decide that every complaint involving a credit card transaction will be investigated. That's an easy rule to follow. Don't pick a rule that is ambiguous.

At first, the temptation may be to make radical and deep changes to every billing system and process. Don't. Instead, keep the meetings short and pick relatively simple changes at each of the five levels of the inquiry. Over time, as the team gets more comfortable with the process, you can expand it to include more and more types of billing complaints and then to other kinds of problems.

### Appoint a Five Whys Master

To facilitate learning, I have found it helpful to appoint a Five Whys master for each area in which the method is being used. This individual is tasked with being the moderator for each Five Whys meeting, making decisions about which prevention steps to take, and assigning the follow-up work from that meeting. The master must be senior enough to have the authority to ensure that those assignments get done but should not be so senior that he or she will not be able to be present at the meetings because of conflicting responsibilities. The Five Whys master is the point person in terms of accountability; he or she is the primary change agent. People in this position can assess how well the meetings are going and whether the prevention investments that are being made are paying off.

### THE FIVE WHYS IN ACTION

IGN Entertainment, a division of News Corporation, is an online video games media company with the biggest audience of video

game players in the world. More than 45 million gamers frequent its portfolio of media properties. IGN was founded in the late 1990s, and News Corporation acquired it in 2005. IGN has grown to employ several hundred people, including almost a hundred engineers.

Recently, I had the opportunity to speak to the product development team at IGN. They had been successful in recent years, but like all the established companies we've seen throughout this book, they were looking to accelerate new product development and find ways to be more innovative. They brought together their engineering, product, and design teams to talk through ways they could apply the Lean Startup model.

This change initiative had the support of IGN's senior management, including the CEO, the head of product development, the vice president of engineering, the publisher, and the head of product. Their previous efforts at Five Whys had not gone smoothly. They had attempted to tackle a laundry list of problem areas nominated by the product team. The issues varied from discrepancies in web analytics to partner data feeds that were not working. Their first Five Whys meeting took an hour, and although they came up with some interesting takeaways, as far as the Five Whys goes, it was a disaster. None of the people who were connected to and knew the most about the issues were at the meeting, and because this was the first time they were doing the Five Whys together, they didn't stick to the format and went off on many tangents. It wasn't a complete waste of time, but it didn't have any of the benefits of the adaptive style of management discussed in this chapter.

## Don't Send Your Baggage through the Five Whys Process

IGN had the experience of trying to solve all of its "baggage" issues that had been causing wasted time for many years. Because this is an overwhelming set of problems, finding fixes quickly proves overwhelming.

In their zeal to get started with the Five Whys, IGN neglected three important things:

1. To introduce Five Whys to an organization, it is necessary to hold Five Whys sessions as new problems come up. Since baggage issues are endemic, they naturally come up as part of the Five Whys analysis and you can take that opportunity to fix them incrementally. If they don't come up organically, maybe they're not as big as they seem.
2. Everyone who is connected to a problem needs to be at the Five Whys session. Many organizations face the temptation to save time by sparing busy people from the root cause analysis. This is a false economy, as IGN discovered the hard way.
3. At the beginning of each Five Whys session, take a few minutes to explain what the process is for and how it works for the benefit of those who are new to it. If possible, use an example of a successful Five Whys session from the past. If you're brand new, you can use my earlier example about the manager who doesn't believe in training. IGN learned that, whenever possible, it helps to use something that has personal meaning for the team.

After our meeting, the IGN leadership decided to give Five Whys another try. Following the advice laid out in this chapter, they appointed a Five Whys master named Tony Ford, a director of engineering. Tony was an entrepreneur who had come to IGN through an acquisition. He got his start with Internet technology, building websites about video games in the late 1990s. Eventually that led to an opportunity at a startup, TeamXbox, where he served as the lead software developer. TeamXbox was acquired by IGN Entertainment in 2003, and since that time Tony has been a technologist, leader of innovation, and proponent of agile and lean practices there.

Unfortunately, Tony started without picking a narrow problem area on which to focus. This led to early setbacks and frustration.

Tony relates, “As the new master I wasn’t very good at traversing through the Five Whys effectively, and the problems we were trying to solve were not great candidates in the first place. As you can imagine, these early sessions were awkward and in the end not very useful. I was getting quite discouraged and frustrated.” This is a common problem when one tries to tackle too much at once, but it is also a consequence of the fact that these skills take time to master. Luckily, Tony persevered: “Having a Five Whys master is critical in my opinion. Five Whys is easy in theory but difficult in practice, so you need someone who knows it well to shape the sessions for those who don’t.”

The turnaround came when Tony led a Five Whys session involving a project that had been missing its deadlines. The session was fascinating and insightful and produced meaningful proportional investments. Tony explains: “The success had to do with a more experienced master and more experienced attendees. We all knew what the Five Whys was, and I did a really good job keeping us on track and away from tangents. This was a pivotal moment. Right then I knew the Five Whys was a new tool that was going to have a real impact on our overall success as a team and as a business.”

On the surface, Five Whys seems to be about technical problems and preventing mistakes, but as teams drive out these superficial wastes, they develop a new understanding of how to work together. Tony put it this way: “I daresay that I discovered that the Five Whys transcends root cause analysis by revealing information that brings your team closer through a common understanding and perspective. A lot of times a problem can pull people apart; Five Whys does the opposite.”

I asked Tony to provide an example of a recent successful Five Whys analysis from IGN. His account of it is listed in the sidebar.

Why couldn't you add or edit posts on the blogs?

Answer: Any post request (write) to the article content api was

returning a 500 error.

Proportional investment: Jim—We'll work on the API, but let's make our CMS more forgiving for the user. Allow users to add and edit drafts without errors for a better user experience.

Why was the content API returning 500 errors?

Answer: The `bson_ext` gem was incompatible with other gems it depends upon.

Proportional investment: King—Remove the gem (already done to resolve the outage).

Why was the gem incompatible?

Answer: We added a new version of the gem in addition to the existing version and the app started using it unexpectedly.

Proportional investment: Bennett—Convert our rails app to use bundler for gem management.

Why did we add a new version of a gem in production without testing?

Answer: We didn't think we needed a test in these cases.

Proportional investment: Bennett and Jim—Write a unit or functional test in the API and CMS that will catch this in the future.

Why do we add additional gems that we don't intend to use right away?

Answer: In preparation for a code push we wanted to get all new gems ready in the production environment. Even though our code deployments are fully automated, gems are not.

Proportional investment: Bennett—Automate gem management



and installation into Continuous Integration and Continuous Deployment process.

Bonus—Why are we doing things in production on Friday nights?

Answer: Because no one says we can't and it was a convenient time for the developer to prepare for a deployment we'd be doing on Monday.

Proportional investment: Tony—Make an announcement to the team. There will be no production changes on Friday, Saturday, or Sunday unless an exception has been made and approved by David (VP Engineering). We will reevaluate this policy when we have a fully automated continuous deployment process in place.

As a result of this Five Whys session and the proportional investments we made, our deployments are easier, quicker, and never again will our process allow a developer to place gems into production systems with unintended consequences. Indeed, we have not had another issue like this. We strengthened our “cluster immune system” as you would say.

Without the Five Whys, we would have never discovered all of the information we did here. My guess is that we would have told that one developer to not do stupid things on Friday nights and moved on. This is what I emphasized earlier, where a good Five Whys session has two outputs, learning and doing. The proportional investments that came out of this session are obviously valuable, but the learnings are much more subtle, but amazing for growing as developers and as a team.

## ADAPTING TO SMALLER BATCHES

Before leaving the topic of building an adaptive organization, I want to introduce one more story. This one concerns a product that you've probably used if you've ever run your own business. It's called QuickBooks, and it is one of Intuit's flagship products.

QuickBooks has been the leading product in its category for many years. As a result, it has a large and dedicated customer base, and Intuit expects it to contribute significantly to its bottom line. Like most personal computer (PC) software of the last two decades, QuickBooks has been launched on an annual cycle, in one giant batch. This was how it worked three years ago, when Greg Wright, the director of product marketing for QuickBooks, joined the team. As you can imagine, there were lots of existing processes in place to ensure a consistent product and an on-time release. The typical release approach was to spend significant up-front time to identify the customers' need:

Typically the first three to four months of each annual cycle was spent strategizing and planning, without building new features. Once a plan and milestones were established, the team would spend the next six to nine months building. This would culminate in a big launch, and then the team would get its first feedback on whether it had successfully delivered on customers' needs at the end of the process.

So here was the time line: start process in September, first beta release is in June, second beta is in July. The beta is essentially testing to make sure it doesn't crash people's computers or cause them to lose their data—by that time in the process, only major bugs can be fixed. The design of the product itself is locked.

This is the standard “waterfall” development methodology that product development teams have used for years. It is a linear, large-batch system that relies for success on proper forecasting and planning. In other words, it is completely maladapted for today's

rapidly changing business environment.

## Year One: Achieving Failure

Greg witnessed this breakdown in 2009, his first year on the QuickBooks team. That year, the company shipped an entirely new system in QuickBooks for online banking, one of its most important features. The team went through rounds of usability testing using mock-ups and nonfunctional prototypes, followed by significant beta testing using sample customer data. At the moment of the launch, everything looked good.

The first beta release was in June, and customer feedback started coming in negative. Although customers were complaining, there wasn't sufficient cause to stop the release because it was technically flawless—it didn't crash computers. At that point, Greg was in a bind. He had no way of knowing how the feedback would translate to real customer behavior in the market. Were these just isolated complaints, or part of a widespread problem? He did know one thing for sure, though: that his team could not afford to miss the deadline.

When the product finally shipped, the results were terrible. It took customers four to five times longer to reconcile their banking transactions than it had with the older version. In the end, Greg's team had failed to deliver on the customer need they were trying to address (despite building the product to specification), and because the next release had to go through the same waterfall process, it took the team nine months to fix. This is a classic case of “achieving failure”—successfully executing a flawed plan.

Intuit uses a tracking survey called the Net Promoter Score<sup>2</sup> to evaluate customer satisfaction with its many products. This is a great source of actionable metrics about what customers really think about a product. In fact, I used it at IMVU, too. One thing that is nice about NPS is that it is very stable over time. Since it is measuring core customer satisfaction, it is not subject to minor fluctuations: it registers only major changes in customer sentiment.

That year, the QuickBooks score dropped 20 points, the first time the company had ever moved the needle with the Net Promoter Score. That 20-point drop resulted in significant losses for Intuit and was embarrassing for the company—all because customer feedback came too late in the process, allowing no time to iterate.

Intuit's senior management, including the general manager of the small business division and the head of small business accounting, recognized the need for change. To their credit, they tasked Greg with driving that change. His mission: to achieve startup speed for the development and deployment of QuickBooks.

## Year Two: Muscle Memory

The next chapter of this story illustrates how hard it is to build an adaptive organization. Greg set out to change the QuickBooks development process by using four principles:

1. Smaller teams. Shift from large teams with uniform functional roles to smaller, fully engaged teams whose members take on different roles.
2. Achieve shorter cycle times.
3. Faster customer feedback, testing both whether it crashes customers' computers and the performance of new features/customer experience.
4. Enable and empower teams to make fast and courageous decisions.

On the surface, these goals seem to be aligned with the methods and principles described in previous chapters, but Greg's second year with QuickBooks was not a marked success. For example, he decreed that the team would move to a midyear release milestone, effectively cutting the cycle time and batch size in half. However, this was not successful. Through sheer determination, the team tried valiantly to get an alpha release out in January. However, the problems that afflict large-batch development were still present.

and the team struggled to complete the alpha by April. That represented an improvement over the past system because issues could be brought to the surface two months earlier than under the old way, but it did not produce the dramatically better results Greg was looking for.

In fact, over the course of the year, the team's process kept looking more and more like it had in prior years. As Greg put it, "Organizations have muscle memory," and it is hard for people to unlearn old habits. Greg was running up against a system, and making individual changes such as arbitrarily changing the release date were no match for it.

### Year Three: Explosion

Frustrated by the limited progress in the previous year, Greg teamed up with the product development leader Himanshu Baxi. Together they tossed out all the old processes. They made a public declaration that their combined teams would be creating new processes and that they were not going to go back to the old way.

Instead of focusing on new deadlines, Greg and Himanshu invested in process, product, and technology changes that enabled working in smaller batches. Those technical innovations helped them get the desktop product to customers faster for feedback. Instead of building a comprehensive road map at the beginning of the year, Greg kicked off the year with what they called idea/code/solution jams that brought engineers, product managers, and customers together to create a pipeline of ideas. It was scary for Greg as a product manager to start the year without a defined list of what would be in the product release, but he had confidence in his team and the new process.

There were three differences in year three:

- Teams were involved in creating new technologies, processes, and systems.
- Cross-functional teams were formed around new great ideas.

- Customers were involved from the inception of each feature concept.

It's important to understand that the old approach did not lack customer feedback or customer involvement in the planning process. In the true spirit of *genchi gembutsu*, Intuit product managers (PMs) would do “follow-me-homes” with customers to identify problems to solve in the next release. However, the PMs were responsible for all the customer research. They would bring it back to the team and say, “This is the problem we want to solve, and here are ideas for how we could solve it.”

Changing to a cross-functional way of working was not smooth sailing. Some team members were skeptical. For example, some product managers felt that it was a waste of time for engineers to spend time in front of customers. The PMs thought that their job was to figure out the customer issue and define what needed to be built. Thus, the reaction of some PMs to the change was: “What’s my job? What am I supposed to be doing?” Similarly, some on the engineering side just wanted to be told what to do; they didn’t want to talk to customers. As is typically the case in large-batch development, both groups had been willing to sacrifice the team’s ability to learn in order to work more “efficiently.”

Communication was critical for this change process to succeed. All the team leaders were open about the change they were driving and why they were driving it. Much of the skepticism they faced was based on the fact that they did not have concrete examples of where this had worked in the past; it was an entirely new process for Intuit. They had to explain clearly why the old process didn’t work and why the annual release “train” was not setting them up for success. Throughout the change they communicated the process outcomes they were shooting for: earlier customer feedback and a faster development cycle that was decoupled from the annual release time line. They repeatedly emphasized that the new approach was how startup competitors were working and iterating. They had to follow suit or risk becoming irrelevant.

Historically, QuickBooks had been built with large teams and long cycle times. For example, in earlier years the ill-fated online banking team had been composed of fifteen engineers, seven quality assurance specialists, a product manager, and at times more than one designer. Now no team is bigger than five people. The focus of each team is iterating with customers as rapidly as possible, running experiments, and then using validated learning to make real-time investment decisions about what to work on. As a result, whereas they used to have five major “branches” of QuickBooks that merged features at the time of the launch, now there are twenty to twenty-five branches. This allows for a much larger set of experiments. Each team works on a new feature for approximately six weeks end to end, testing it with real customers throughout the process.

Although the primary changes that are required in an adaptive organization are in the mind-set of its employees, changing the culture is not sufficient. As we saw in [Chapter 9](#), lean management requires treating work as a system and then dealing with the batch size and cycle time of the whole process. Thus, to achieve lasting change, the QuickBooks team had to invest in tools and platform changes that would enable the new, faster way of working.

For example, one of the major stress points in the attempt to release an early alpha version the previous year was that QuickBooks is a mission-critical product. Many small businesses use it as their primary repository for critical financial data. The team was extremely wary of releasing a minimum viable product that had any risk of corrupting customer data. Therefore, even if they worked in smaller teams with a smaller scope, the burden of all that risk would have made it hard to work in smaller batches.

To get the batch size down, the QuickBooks team had to invest in new technology. They built a virtualization system that allowed them to run multiple versions of QuickBooks on a customer’s computer. The second version could access all the customer’s data but could not make permanent changes to it. Thus, there was no risk of the new version corrupting the customer’s data by accident.

This allowed them to isolate new releases to allow selected real customers to test them and provide feedback.

The results in year three were promising. The version of QuickBooks that shipped that year had significantly higher customer satisfaction ratings and sold more units. If you're using QuickBooks right now, odds are you are using a version that was built in small batches. As Greg heads into his fourth year with the QuickBooks team, they are exploring even more ways to drive down batch size and cycle time. As usual, there are possibilities that go beyond technical solutions. For example, the annual sales cycle of boxed desktop software is a significant barrier to truly rapid learning, and so the team has begun experimenting with subscription-based products for the most active customers. With customers downloading updates online, Intuit can release software on a more frequent basis. Soon this program will see the QuickBooks team releasing to customers quarterly.<sup>3</sup>



As Lean Startups grow, they can use adaptive techniques to develop more complex processes without giving up their core advantage: speed through the Build-Measure-Learn feedback loop. In fact, one of the primary benefits of using techniques that are derived from lean manufacturing is that Lean Startups, when they grow up, are well positioned to develop operational excellence based on lean principles. They already know how to operate with discipline, develop processes that are tailor-made to their situation, and use lean techniques such as the Five Whys and small batches. As a successful startup makes the transition to an established company, it will be well poised to develop the kind of culture of disciplined execution that characterizes the world's best firms, such as Toyota.

However, successfully growing into an established company is not the end of the story. A startup's work is never done, because as was discussed in [Chapter 2](#), even established companies must struggle to find new sources of growth through disruptive innovation. This imperative is coming earlier in companies' lives. No longer can a



successful startup expect to have years after its initial public offering to bask in market-leading success. Today successful companies face immediate pressure from new competitors, fast followers, and scrappy startups. As a result, it no longer makes sense to think of startups as going through discrete phases like the proverbial metamorphosis of a caterpillar to a butterfly. Both successful startups and established companies alike must learn to juggle multiple kinds of work at the same time, pursuing operational excellence and disruptive innovation. This requires a new kind of portfolio thinking, which is the subject of [Chapter 12](#).

# 12

## INNOVATE

Conventional wisdom holds that when companies become larger, they inevitably lose the capacity for innovation, creativity, and growth. I believe this is wrong. As startups grow, entrepreneurs can build organizations that learn how to balance the needs of existing customers with the challenges of finding new customers to serve, managing existing lines of business, and exploring new business models—all at the same time. And, if they are willing to change their management philosophy, I believe even large, established companies can make this shift to what I call portfolio thinking.

### HOW TO NURTURE DISRUPTIVE INNOVATION

Successful innovation teams must be structured correctly in order to succeed. Venture-backed and bootstrapped startups naturally have some of these structural attributes as a consequence of being small, independent companies. Internal startup teams require support from senior management to create these structures. Internal or external, in my experience startup teams require three structural attributes: scarce but secure resources, independent authority to develop their business, and a personal stake in the outcome. Each of these requirements is different from those of established company divisions. Keep in mind that structure is merely a prerequisite—it does not guarantee success. But getting the structure wrong can lead to almost certain failure.

## Scarce but Secure Resources

Division leaders in large, established organizations are adept at using politics to enlarge their budgets but know that those budgets are somewhat loose. They often acquire as large a budget as possible and prepare to defend it against incursions from other departments. Politics means that they sometimes win and sometimes lose: if a crisis emerges elsewhere in the organization, their budget might suddenly be reduced by 10 percent. This is not a catastrophe; teams will have to work harder and do more with less. Most likely, the budget has some padding in anticipation of this kind of eventuality.

Startups are different: too much budget is as harmful as too little—as countless dot-com failures can attest—and startups are extremely sensitive to midcourse budgetary changes. It is extremely rare for a stand-alone startup company to lose 10 percent of its cash on hand suddenly. In a large number of cases, this would be a fatal blow, as independent startups are run with little margin for error. Thus, startups are both easier and more demanding to run than traditional divisions: they require much less capital overall, but that capital must be absolutely secure from tampering.

## Independent Development Authority

Startup teams need complete autonomy to develop and market new products within their limited mandate. They have to be able to conceive and execute experiments without having to gain an excessive number of approvals.

I strongly recommend that startup teams be completely cross-functional, that is, have full-time representation from every functional department in the company that will be involved in the creation or launch of their early products. They have to be able to build and ship actual functioning products and services, not just prototypes. Handoffs and approvals slow down the Build-Measure-

Learn feedback loop and inhibit both learning and accountability. Startups require that they be kept to an absolute minimum.

Of course, this level of development autonomy is liable to raise fears in a parent organization. Alleviating those fears is a major goal of the method recommended below.

## A Personal Stake in the Outcome

Third, entrepreneurs need a personal stake in the outcome of their creations. In stand-alone new ventures, this usually is achieved through stock options or other forms of equity ownership. Where a bonus system must be used instead, the best incentives are tied to the long-term performance of the new innovation.

However, I do not believe that a personal stake has to be financial. This is especially important in organizations, such as nonprofits and government, in which the innovation is not tied to financial objectives. In these cases, it is still possible for teams to have a personal stake. The parent organization has to make it clear who the innovator is and make sure the innovator receives credit for having brought the new product to life—if it is successful. As one entrepreneur who ran her own division at a major media company told me, “Financial incentives aside, I always felt that because my name was on the door, I had more to lose and more to prove than someone else. That sense of ownership is not insignificant.”

This formula is effective in for-profit companies as well. At Toyota, the manager in charge of developing a new vehicle from start to finish is called the shusa, or chief engineer:

Shusa are often called heavy-weight project managers in the U.S. literature, but this name understates their real roles as design leaders. Toyota employees translate the term as chief engineer, and they refer to the vehicle under development as the shusa's car. They assured us that the shusa has final, absolute authority over every aspect of vehicle

development.<sup>1</sup>

On the flip side, I know an extremely high-profile technology company that has a reputation for having an innovative culture, yet its track record of producing new products is disappointing. The company boasts an internal reward system that is based on large financial and status awards to teams that do something extraordinary, but those awards are handed out by senior management on the basis of—no one knows what. There are no objective criteria by which a team can gauge whether it will win this coveted lottery. Teams have little confidence that they will receive any long-term ownership of their innovations. Thus, teams rarely are motivated to take real risks, instead focusing their energies on projects that are expected to win the approval of senior management.

## CREATING A PLATFORM FOR EXPERIMENTATION

Next, it is important to focus on establishing the ground rules under which autonomous startup teams operate: how to protect the parent organization, how to hold entrepreneurial managers accountable, and how to reintegrate an innovation back into the parent organization if it is successful. Recall the “island of freedom” that enabled the SnapTax team—in [Chapter 2](#)—to successfully create a startup within Intuit. That’s what a platform for experimentation can do.

### Protecting the Parent Organization

Conventionally, advice about internal innovators focuses on protecting the startup from the parent organization. I believe it is necessary to turn this model on its head.

Let me begin by describing a fairly typical meeting from one of my consulting clients, a large company. Senior management had gathered to make decisions about what to include in the next

version of its product. As part of the company's commitment to being data-driven, it had tried to conduct an experiment on pricing. The first part of the meeting was taken up with interpreting the data from the experiment.

One problem was that nobody could agree on what the data meant. Many custom reports had been created for the meeting; the data warehouse team was at the meeting too. The more they were asked to explain the details of each row on the spreadsheet, the more evident it became that nobody understood how those numbers had been derived. What we were left looking at was the number of gross sales of the product at a variety of different price points, broken down by quarter and by customer segment. It was a lot of data to try to comprehend.

Worse, nobody was sure which customers had been exposed to the experiment. Different teams had been responsible for implementing it, and so different parts of the product had been updated at different times. The whole process had taken many months, and by this point, the people who had conceived the experiment had been moved to a division separate from that of the people who had executed it.

You should be able to spot the many problems with this situation: the use of vanity metrics instead of actionable metrics, an overly long cycle time, the use of large batch sizes, an unclear growth hypothesis, a weak experimental design, a lack of team ownership, and therefore very little learning.

Listening in, I assumed this would be the end of the meeting. With no agreed-on facts to help make the decision, I thought nobody would have any basis for making the case for a particular action. I was wrong. Each department simply took whatever interpretation of the data supported its position best and started advocating on its own behalf. Other departments would chime in with alternative interpretations that supported their positions, and so on. In the end, decisions were not made based on data. Instead, the executive running the meeting was forced to base decisions on the most plausible-sounding arguments.

It seemed wasteful to me how much of the meeting had been

spent debating the data because, in the end, the arguments that carried the day could have been made right at the start. It was as if each advocate sensed that he or she was about to be ambushed; if another team managed to bring clarity to the situation, it might undermine that person, and so the rational response was to obfuscate as much as possible. What a waste.

Ironically, meetings like this had given data-driven decision making and experimentation a bad name inside the company, and for good reason. The data warehousing team was producing reports that nobody read or understood. The project teams felt the experiments were a waste of time, since they involved building features halfway, which meant they were never any good. “Running an experiment” seemed to them to be code for postponing a hard decision. Worst of all, the executive team experienced the meetings as chronic headaches. Their old product prioritization meetings might have been little more than a battle of opinions, but at least the executives understood what was going on. Now they had to go through a ritual that involved complex math and reached no definite outcome, and then they ended up having a battle of opinions anyway.

## Rational Fears

However, at the heart of this departmental feud was a very rational fear. This company served two customer segments: a business-to-business enterprise segment and a consumer segment. In the B2B segment, the company employed sales staff to sell large volumes of the product to other companies, whereas the consumer segment was driven mostly by one-off purchases made by individuals. The bulk of the company’s current revenue came from B2B sales, but growth in that segment had been slowing. Everyone agreed there was tremendous potential for growth in the consumer segment, but so far little had materialized.

Part of the cause of this lack of growth was the current pricing structure. Like many companies that sell to large enterprises, this

one published a high list price and then provided heavy discounts to “favored” corporate clients who bought in bulk. Naturally, every salesperson was encouraged to make all of his or her clients feel favored. Unfortunately, the published list price was much too high for the consumer segment.

The team in charge of growing the consumer segment wanted to run experiments with a lower price structure. The team in charge of the enterprise segment was nervous that this would cannibalize or otherwise diminish its existing relationships with its customers. What if those customers discovered that individuals were getting a lower price than they were?

Anyone who has been in a multisegment business will recognize that there are many possible solutions to this problem, such as creating tiered feature sets so that different customers are able to purchase different “levels” of the product (as in airline seating) or even supporting different products under separate brand names. Yet the company was struggling to implement any of those solutions. Why? Out of fear of endangering the current business, each proposed experiment would be delayed, sabotaged, and obfuscated.

It's important to emphasize that this fear is well founded. Sabotage is a rational response from managers whose territory is threatened. This company is not a random, tiny startup with nothing to lose. An established company has a lot to lose. If the revenue from the core business goes down, heads will roll. This is not something to be taken lightly.

## The Dangers of Hiding Innovation inside the Black Box

The imperative to innovate is unrelenting. Without the ability to experiment in a more agile manner, this company eventually would suffer the fate described in The Innovator's Dilemma: ever-higher profits and margins year after year until the business suddenly collapsed.

We often frame internal innovation challenges by asking, How can we protect the internal startup from the parent organization? I



would like to reframe and reverse the question: How can we protect the parent organization from the startup? In my experience, people defend themselves when they feel threatened, and no innovation can flourish if defensiveness is given free rein. In fact, this is why the common suggestion to hide the innovation team is misguided. There are examples of one-time successes using a secret skunkworks or off-site innovation team, such as the building of the original IBM PC in Boca Raton, Florida, completely separate from mainline IBM. But these examples should serve mostly as cautionary tales, because they have rarely led to sustainable innovation.<sup>2</sup> Hiding from the parent organization can have long-term negative consequences.

Consider it from the point of view of the managers who have the innovation sprung on them. They are likely to feel betrayed and more than a little paranoid. After all, if something of this magnitude could be hidden, what else is waiting in the shadows? Over time, this leads to more politics as managers are incentivized to ferret out threats to their power, influence, and careers. The fact that the innovation was a success is no justification for this dishonest behavior. From the point of view of established managers, the message is clear: if you are not on the inside, you are liable to be blindsided by this type of secret.

It is unfair to criticize these managers for their response; the criticism should be aimed at senior executives who failed to design a supportive system in which to operate and innovate. I believe this is one reason why companies such as IBM lost their leadership position in the new markets that they developed using a black box such as the PC business; they are unable to re-create and sustain the culture that led to the innovation in the first place.

## Creating an Innovation Sandbox

The challenge here is to create a mechanism for empowering innovation teams out in the open. This is the path toward a sustainable culture of innovation over time as companies face

repeated existential threats. My suggested solution is to create a sandbox for innovation that will contain the impact of the new innovation but not constrain the methods of the startup team. It works as follows:

1. Any team can create a true split-test experiment that affects only the sandboxed parts of the product or service (for a multipart product) or only certain customer segments or territories (for a new product). However:
2. One team must see the whole experiment through from end to end.
3. No experiment can run longer than a specified amount of time (usually a few weeks for simple feature experiments, longer for more disruptive innovations).
4. No experiment can affect more than a specified number of customers (usually expressed as a percentage of the company's total mainstream customer base).
5. Every experiment has to be evaluated on the basis of a single standard report of five to ten (no more) actionable metrics.
6. Every team that works inside the sandbox and every product that is built must use the same metrics to evaluate success.
7. Any team that creates an experiment must monitor the metrics and customer reactions (support calls, social media reaction, forum threads, etc.) while the experiment is in progress and abort it if something catastrophic happens.

At the beginning, the sandbox has to be quite small. In the company above, the sandbox initially contained only the pricing page. Depending on the types of products the company makes, the size of the sandbox can be defined in different ways. For example, an online service might restrict it to certain pages or user flows. A retail operation might restrict it to certain stores or geographic areas. Companies trying to bring an entirely new product to market might build the restriction around customers in certain segments.

Unlike in a concept test or market test, customers in the sandbox

are considered real and the innovation team is allowed to attempt to establish a long-term relationship with them. After all, they may be experimenting with those early adopters for a long time before their learning milestones are accomplished.

Whenever possible, the innovation team should be cross-functional and have a clear team leader, like the Toyota shusa. It should be empowered to build, market, and deploy products or features in the sandbox without prior approval. It should be required to report on the success or failure of those efforts by using standard actionable metrics and innovation accounting.

This approach can work even for teams that have never before worked cross-functionally. The first few changes, such as a price change, may not require great engineering effort, but they require coordination across departments: engineering, marketing, customer service. Teams that work this way are more productive as long as productivity is measured by their ability to create customer value and not just stay busy.

True experiments are easy to classify as successes or failures because top-level metrics either move or they don't. Either way, the team learns immediately whether its assumptions about how customers will behave are correct. By using the same metrics each time, the team builds literacy about those metrics across the company. Because the innovation team is reporting on its progress by using the system of innovation accounting described in [Part Two](#), anyone who reads those reports is getting an implicit lesson in the power of actionable metrics. This effect is extremely powerful. Even if someone wants to sabotage the innovation team, he or she will have to learn all about actionable metrics and learning milestones to do it.

The sandbox also promotes rapid iteration. When people have a chance to see a project through from end to end and the work is done in small batches and delivers a clear verdict quickly, they benefit from the power of feedback. Each time they fail to move the numbers, they have a real opportunity to act on their findings immediately. Thus, these teams tend to converge on optimal solutions rapidly even if they start out with really bad ideas.

As we saw earlier, this is a manifestation of the principle of small batches. Functional specialists, especially those steeped in waterfall or stage-gate development, have been trained to work in extremely large batches. This causes even good ideas to get bogged down by waste. By making the batch size small, the sandbox method allows teams to make cheap mistakes quickly and start learning. As we'll see below, these small initial experiments can demonstrate that a team has a viable new business that can be integrated back into the parent company.

## Holding Internal Teams Accountable

We already discussed learning milestones in detail in [Chapter 7](#). With an internal startup team, the sequence of accountability is the same: build an ideal model of the desired disruption that is based on customer archetypes, launch a minimum viable product to establish a baseline, and then attempt to tune the engine to get it closer to the ideal.

Operating in this framework, internal teams essentially act as startups. As they demonstrate success, they need to become integrated into the company's overall portfolio of products and services.

## CULTIVATING THE MANAGEMENT PORTFOLIO

There are four major kinds of work that companies must manage.<sup>3</sup> As an internal startup grows, the entrepreneurs who created the original concept must tackle the challenge of scale. As new mainstream customers are acquired and new markets are conquered, the product becomes part of the public face of the company, with important implications for PR, marketing, sales, and business development. In most cases, the product will attract competitors: copycats, fast followers, and imitators of all stripes.

Once the market for the new product is well established,

procedures become more routine. To combat the inevitable commoditization of the product in its market, line extensions, incremental upgrades, and new forms of marketing are essential. In this phase, operational excellence takes on a greater role, as an important way to increase margins is to lower costs. This may require a different type of manager: one who excels in optimization, delegation, control, and execution. Company stock prices depend on this kind of predictable growth.

There is a fourth phase as well, one dominated by operating costs and legacy products. This is the domain of outsourcing, automation, and cost reduction. Nonetheless, infrastructure is still mission-critical. Failure of facilities or important infrastructure or the abandonment of loyal customers could derail the whole company. However, unlike the growth and optimization phase, investments in this area will not help the company achieve top-line growth. Managers of this kind of organization suffer the fate of baseball umpires: criticized when something goes wrong, unappreciated when things are going well.

We tend to speak of these four phases of businesses from the perspective of large companies, in which they may represent entire divisions and hundreds or even thousands of people. That's logical, as the evolution of the business in these kinds of extreme cases is the easiest to observe. However, all companies engage in all four phases of work all the time. As soon as a product hits the marketplace, teams of people work hard to advance it to the next phase. Every successful product or feature began life in research and development (R&D), eventually became a part of the company's strategy, was subject to optimization, and in time became old news.

The problem for startups and large companies alike is that employees often follow the products they develop as they move from phase to phase. A common practice is for the inventor of a new product or feature to manage the subsequent resources, team, or division that ultimately commercializes it. As a result, strong creative managers wind up getting stuck working on the growth and optimization of products rather than creating new ones.

This tendency is one of the reasons established companies

struggle to find creative managers to foster innovation in the first place. Every new innovation competes for resources with established projects, and one of the scarcest resources is talent.

## Entrepreneur Is a Job Title

The way out of this dilemma is to manage the four kinds of work differently, allowing strong cross-functional teams to develop around each area. When products move from phase to phase, they are handed off between teams. Employees can choose to move with the product as part of the handoff or stay behind and begin work on something new. Neither choice is necessarily right or wrong; it depends on the temperament and skills of the person in question.

Some people are natural inventors who prefer to work without the pressure and expectations of the later business phases. Others are ambitious and see innovation as a path toward senior management. Still others are particularly skilled at the management of running an established business, outsourcing, and bolstering efficiencies and wringing out cost reductions. People should be allowed to find the kinds of jobs that suit them best.

In fact, entrepreneurship should be considered a viable career path for innovators inside large organizations. Managers who can lead teams by using the Lean Startup methodology should not have to leave the company to reap the rewards of their skills or have to pretend to fit into the rigid hierarchies of established functional departments. Instead, they should have a business card that says simply “Entrepreneur” under the name. They should be held accountable via the system of innovation accounting and promoted and rewarded accordingly.

After an entrepreneur has incubated a product in the innovation sandbox, it has to be reintegrated into the parent organization. A larger team eventually will be needed to grow it, commercialize it, and scale it. At first, this team will require the continued leadership of the innovators who worked in the sandbox. In fact, this is a positive part of the process in that it gives the innovators a chance

to train new team members in the new style of working that they mastered in the original sandbox.

Ideally, the sandbox will grow over time; that is, rather than move the team out of the sandbox and into the company's standard routines, there may be opportunities to enlarge the scope of the sandbox. For example, if only certain aspects of the product were subject to experimentation in the sandbox, new features can be added. In the online service described earlier, this could be accomplished by starting with a sandbox that encompassed the product pricing page. When those experiments succeeded, the company could add the website's home page to the sandbox. It subsequently might add the search functionality or the overall web design. If only certain customers or certain numbers of customers were targeted initially, the product's reach could be increased. When such changes are contemplated, it's important that senior management consider whether the teams working in the sandbox can fend for themselves politically in the parent organization. The sandbox was designed to protect them and the parent organization, and any expansion needs to take this into account.

Working in the innovation sandbox is like developing startup muscles. At first, the team will be able to take on only modest experiments. The earliest experiments may fail to produce much learning and may not lead to scalable success. Over time, those teams are almost guaranteed to improve as long as they get the constant feedback of small-batch development and actionable metrics and are held accountable to learning milestones.

Of course, any innovation system eventually will become the victim of its own success. As the sandbox expands and the company's revenue grows as a result of the sandbox's innovations, the cycle will have to begin again. The former innovators will become guardians of the status quo. When the product makes up the whole sandbox, it inevitably will become encumbered with the additional rules and controls needed for mission-critical operation. New innovation teams will need a new sandbox within which to play.

## Becoming the Status Quo

This last transition is especially hard for innovators to accept: their transformation from radical outsiders to the embodiment of the status quo. I have found it disturbing in my career. As you can guess from the techniques I advocate as part of the Lean Startup, I have always been a bit of a troublemaker at the companies at which I have worked, pushing for rapid iteration, data-driven decision making, and early customer involvement. When these ideas were not part of the dominant culture, it was simple (if frustrating) to be an advocate. All I had to do was push as hard as humanly possible for my ideas. Since the dominant culture found them heretical, they would compromise with me a “reasonable” amount. Thanks to the psychological phenomenon of anchoring, this led to a perverse incentive: the more radical my suggestion was, the more likely it was that the reasonable compromise would be closer to my true goal.

Fast-forward several years to when I was running product development. When we’d hire new people, they had to be indoctrinated into the Lean Startup culture. Split testing, continuous deployment, and customer testing were all standard practice. I needed to continue to be a strong advocate for my ideas, making sure each new employee was ready to give them a try. But for the people who had been working there awhile, those ideas had become part of the status quo.

Like many entrepreneurs, I was caught between constant evangelizing for my ideas and constantly entertaining suggestions for ways they could be improved. My employees faced the same incentive I had exploited years before: the more radical the suggestion is, the more likely it is that the compromise will move in the direction they desire. I heard it all: suggestions that we go back to waterfall development, use more quality assurance (QA), use less QA, have more or less customer involvement, use more vision and less data, or interpret data in a more statistically rigorous way.



It took a constant effort to consider these suggestions seriously. However, responding dogmatically is unhelpful. Compromising by automatically splitting the difference doesn't work either.

I've found that every suggestion should be subjected to the same rigorous scientific inquiry that led to the creation of the Lean Startup in the first place. Can we use the theory to predict the results of the proposed change? Can we incubate the change in a small team and see what happens? Can we measure its impact? Whenever they could be implemented, these approaches have allowed me to increase my own learning and, more important, the productivity of the companies I have worked with. Many of the Lean Startup techniques that we pioneered at IMVU are not my original contributions. Rather, they were conceived, incubated, and executed by employees who brought their own creativity and talent to the task.

Above all, I faced this common question: How do we know that "your way" of building a company will work? What other companies are using it? Who has become rich and famous as a result? These questions are sensible. The titans of our industry are all working in a slower, more linear way. Why are we doing something different?

It is these questions that require the use of theory to answer. Those who look to adopt the Lean Startup as a defined set of steps or tactics will not succeed. I had to learn this the hard way. In a startup situation, things constantly go wrong. When that happens, we face the age-old dilemma summarized by Deming: How do we know that the problem is due to a special cause versus a systemic cause? If we're in the middle of adopting a new way of working, the temptation will always be to blame the new system for the problems that arise. Sometimes that tendency is correct, sometimes not. Learning to tell the difference requires theory. You have to be able to predict the outcome of the changes you make to tell if the problems that result are really problems.

For example, changing the definition of productivity for a team from functional excellence—excellence in marketing, sales, or product development—to validated learning will cause problems.

As was indicated earlier, functional specialists are accustomed to measuring their efficiency by looking at the proportion of time they are busy doing their work. A programmer expects to be coding all day long, for example. That is why many traditional work environments frustrate these experts: the constant interruption of meetings, cross-functional handoffs, and explanations for endless numbers of bosses all act as a drag on efficiency. However, the individual efficiency of these specialists is not the goal in a Lean Startup. Instead, we want to force teams to work cross-functionally to achieve validated learning. Many of the techniques for doing this—actionable metrics, continuous deployment, and the overall Build-Measure-Learn feedback loop—necessarily cause teams to suboptimize for their individual functions. It does not matter how fast we can build. It does not matter how fast we can measure. What matters is how fast we can get through the entire loop.

In my years teaching this system, I have noticed this pattern every time: switching to validated learning feels worse before it feels better. That's the case because the problems caused by the old system tend to be intangible, whereas the problems of the new system are all too tangible. Having the benefit of theory is the antidote to these challenges. If it is known that this loss of productivity is an inevitable part of the transition, it can be managed actively. Expectations can be set up front. In my consulting practice, for example, I have learned to raise these issues from day one; otherwise, they are liable to derail the whole effort once it is under way. As the change progresses, we can use the root cause analysis and fast response techniques to figure out which problems need prevention. Ultimately, the Lean Startup is a framework, not a blueprint of steps to follow. It is designed to be adapted to the conditions of each specific company. Rather than copy what others have done, techniques such as the Five Whys allow you to build something that is perfectly suited to your company.

The best way to achieve mastery of and explore these ideas is to embed oneself in a community of practice. There is a thriving community of Lean Startup meetups around the world as well as

online, and suggestions for how you can take advantage of these resources listed in the last chapter of this book, “Join the Movement.”

## EPILOGUE: WASTE NOT

This year marks the one hundredth anniversary of Frederick Winslow Taylor's *The Principles of Scientific Management*, first published in 1911. The movement for scientific management changed the course of the twentieth century by making possible the tremendous prosperity that we take for granted today. Taylor effectively invented what we now consider simply management: improving the efficiency of individual workers, management by exception (focusing only on unexpectedly good or bad results), standardizing work into tasks, the task-plus-bonus system of compensation, and—above all—the idea that work can be studied and improved through conscious effort. Taylor invented modern white-collar work that sees companies as systems that must be managed at more than the level of the individual. There is a reason all past management revolutions have been led by engineers: management is human systems engineering.

In 1911 Taylor wrote: “In the past, the man has been first; in the future, the system must be first.” Taylor's prediction has come to pass. We are living in the world he imagined. And yet, the revolution that he unleashed has been—in many ways—too successful. Whereas Taylor preached science as a way of thinking, many people confused his message with the rigid techniques he advocated: time and motion studies, the differential piece-rate system, and—most galling of all—the idea that workers should be treated as little more than automatons. Many of these ideas proved extremely harmful and required the efforts of later theorists and

managers to undo. Critically, lean manufacturing rediscovered the wisdom and initiative hidden in every factory worker and redirected Taylor's notion of efficiency away from the individual task and toward the corporate organism as a whole. But each of these subsequent revolutions has embraced Taylor's core idea that work can be studied scientifically and can be improved through a rigorous experimental approach.

In the twenty-first century, we face a new set of problems that Taylor could not have imagined. Our productive capacity greatly exceeds our ability to know what to build. Although there was a tremendous amount of invention and innovation in the early twentieth century, most of it was devoted to increasing the productivity of workers and machines in order to feed, clothe, and house the world's population. Although that project is still incomplete, as the millions who live in poverty can attest, the solution to that problem is now strictly a political one. We have the capacity to build almost anything we can imagine. The big question of our time is not Can it be built? but Should it be built? This places us in an unusual historical moment: our future prosperity depends on the quality of our collective imaginations.

In 1911, Taylor wrote:

We can see our forests vanishing, our water-powers going to waste, our soil being carried by floods into the sea; and the end of our coal and our iron is in sight. But our larger wastes of human effort, which go on every day through such of our acts as are blundering, ill-directed, or inefficient ... are less visible, less tangible, and are but vaguely appreciated.

We can see and feel the waste of material things. Awkward, inefficient, or ill-directed movements of men, however, leave nothing visible or tangible behind them. Their appreciation calls for an act of memory, an effort of the imagination. And for this reason, even though our daily loss from this source is greater than from our waste of material things, the one has stirred us deeply, while the

other has moved us but little.<sup>1</sup>

A century on, what can we say about those words? On the one hand, they feel archaic. We of the twenty-first century are hyperaware of the importance of efficiency and the economic value of productivity gains. Our workplaces are—at least when it comes to the building of material objects—incredibly well organized compared with those of Taylor's day.

On the other hand, Taylor's words strike me as completely contemporary. For all of our vaunted efficiency in the making of things, our economy is still incredibly wasteful. This waste comes not from the inefficient organization of work but rather from working on the wrong things—and on an industrial scale. As Peter Drucker said, "There is surely nothing quite so useless as doing with great efficiency what should not be done at all."<sup>2</sup>

And yet we are doing the wrong things efficiently all the time. It is hard to come by a solid estimate of just how wasteful modern work is, but there is no shortage of anecdotes. In my consulting and travels talking about the Lean Startup, I hear the same message consistently from employees of companies big and small. In every industry we see endless stories of failed launches, ill-conceived projects, and large-batch death spirals. I consider this misuse of people's time a criminally negligent waste of human creativity and potential.

What percentage of all this waste is preventable? I think a much larger proportion than we currently realize. Most people I meet believe that in their industry at least, projects fail for good reasons: projects are inherently risky, market conditions are unpredictable, "big company people" are intrinsically uncreative. Some believe that if we just slowed everything down and used a more careful process, we could reduce the failure rate by doing fewer projects of higher quality. Others believe that certain people have an innate gift of knowing the right thing to build. If we can find enough of these visionaries and virtuosos, our problems will be solved. These "solutions" were once considered state of the art in the nineteenth century. too. before people knew about modern management.

The requirements of an ever-faster world make these antique approaches unworkable, and so the blame for failed projects and businesses often is heaped on senior management, which is asked to do the impossible. Alternatively, the finger of blame is pointed at financial investors or the public markets for overemphasizing quick fixes and short-term results. We have plenty of blame to go around, but far too little theory to guide the actions of leaders and investors alike.

The Lean Startup movement stands in contrast to this hand-wringing. We believe that most forms of waste in innovation are preventable once their causes are understood. All that is required is that we change our collective mind-set concerning how this work is to be done.

It is insufficient to exhort workers to try harder. Our current problems are caused by trying too hard—at the wrong things. By focusing on functional efficiency, we lose sight of the real goal of innovation: to learn that which is currently unknown. As Deming taught, what matters is not setting quantitative goals but fixing the method by which those goals are attained. The Lean Startup movement stands for the principle that the scientific method can be brought to bear to answer the most pressing innovation question: How can we build a sustainable organization around a new set of products or services?

## ORGANIZATIONAL SUPERPOWERS

A participant at one of my workshops came up to me a few months afterward to relate the following story, which I am paraphrasing: “Knowing Lean Startup principles makes me feel like I have superpowers. Even though I’m just a junior employee, when I meet with corporate VPs and GMs in my large company, I ask them simple questions and very quickly help them see how their projects are based on fundamental hypotheses that are testable. In minutes, I can lay out a plan they could follow to scientifically validate their plans before it’s too late. They consistently respond with ‘Wow, you

are brilliant. We've never thought to apply that level of rigor to our thinking about new products before.' ”

As a result of these interactions, he has developed a reputation within his large company as a brilliant employee. This has been good for his career but very frustrating for him personally. Why? Because although he is quite brilliant, his insights into flawed product plans are due not to his special intelligence but to having a theory that allows him to predict what will happen and propose alternatives. He is frustrated because the managers he is pitching his ideas to do not see the system. They wrongly conclude that the key to success is finding brilliant people like him to put on their teams. They are failing to see the opportunity he is really presenting them: to achieve better results systematically by changing their beliefs about how innovation happens.

### Putting the System First: Some Dangers

Like Taylor before us, our challenge is to persuade the managers of modern corporations to put the system first. However, Taylorism should act as a cautionary tale, and it is important to learn the lessons of history as we bring these new ideas to a more mainstream audience.

Taylor is remembered for his focus on systematic practice rather than individual brilliance. Here is the full quote from *The Principles of Scientific Management* that includes the famous line about putting the system first:

In the future it will be appreciated that our leaders must be trained right as well as born right, and that no great man can (with the old system of personal management) hope to compete with a number of ordinary men who have been properly organized so as efficiently to cooperate.

In the past the man has been first; in the future the system must be first. This in no sense, however, implies that great men are not needed. On the contrary, the first object of any



good system must be that of developing first-class men; and under systematic management the best man rises to the top more certainly and more rapidly than ever before.<sup>3</sup>

Unfortunately, Taylor's insistence that scientific management does not stand in opposition to finding and promoting the best individuals was quickly forgotten. In fact, the productivity gains to be had through the early scientific management tactics, such as time and motion study, task-plus-bonus, and especially functional foremanship (the forerunner of today's functional departments), were so significant that subsequent generations of managers lost sight of the importance of the people who were implementing them.

This has led to two problems: (1) business systems became overly rigid and thereby failed to take advantage of the adaptability, creativity, and wisdom of individual workers, and (2) there has been an overemphasis on planning, prevention, and procedure, which enable organizations to achieve consistent results in a mostly static world. On the factory floor, these problems have been tackled head on by the lean manufacturing movement, and those lessons have spread throughout many modern corporations. And yet in new product development, entrepreneurship, and innovation work in general we are still using an outdated framework.

My hope is that the Lean Startup movement will not fall into the same reductionist trap. We are just beginning to uncover the rules that govern entrepreneurship, a method that can improve the odds of startup success, and a systematic approach to building new and innovative products. This in no way diminishes the traditional entrepreneurial virtues: the primacy of vision, the willingness to take bold risks, and the courage required in the face of overwhelming odds. Our society needs the creativity and vision of entrepreneurs more than ever. In fact, it is precisely because these are such precious resources that we cannot afford to waste them.

I believe that if Taylor were alive today, he would chuckle at what constitutes the management of entrepreneurs and innovators. Although we harness the labor of scientists and engineers who would have dazzled any early-twentieth-century person with their feats of technical wizardry, the management practices we use to organize them are generally devoid of scientific rigor. In fact, I would go so far as to call them pseudoscience.

We routinely green-light new projects more on the basis of intuition than facts. As we've seen throughout this book, that is not the root cause of the problem. All innovation begins with vision. It's what happens next that is critical. As we've seen, too many innovation teams engage in success theater, selectively finding data that support their vision rather than exposing the elements of the vision to true experiments, or, even worse, staying in stealth mode to create a data-free zone for unlimited "experimentation" that is devoid of customer feedback or external accountability of any kind. Anytime a team attempts to demonstrate cause and effect by placing highlights on a graph of gross metrics, it is engaging in pseudoscience. How do we know that the proposed cause and effect is true? Anytime a team attempts to justify its failures by resorting to learning as an excuse, it is engaged in pseudoscience as well.

If learning has taken place in one iteration cycle, let us demonstrate it by turning it into validated learning in the next cycle. Only by building a model of customer behavior and then showing our ability to use our product or service to change it over time can we establish real facts about the validity of our vision.

Throughout our celebration of the success of the Lean Startup movement, a note of caution is essential. We cannot afford to have our success breed a new pseudoscience around pivots, MVPs, and the like. This was the fate of scientific management, and in the end, I believe, that set back its cause by decades. Science came to stand for the victory of routine work over creative work, mechanization over humanity, and plans over agility. Later movements had to be spawned to correct those deficiencies.

Taylor believed in many things that he dubbed scientific but that our modern eyes perceive as mere prejudice. He believed in the inherent superiority in both intelligence and character of aristocratic men over the working classes and the superiority of men over women; he also thought that lower-status people should be supervised strictly by their betters. These beliefs are part and parcel of Taylor's time, and it is tempting to forgive him for having been blind to them.

Yet when our time is viewed through the lens of future practice, what prejudices will be revealed? In what forces do we place undue faith? What might we risk losing sight of with this initial success of our movement?

It is with these questions that I wish to close. As gratifying as it is for me to see the Lean Startup movement gain fame and recognition, it is far more important that we be right in our prescriptions. What is known so far is just the tip of the iceberg. What is needed is a massive project to discover how to unlock the vast stores of potential that are hidden in plain sight in our modern workforce. If we stopped wasting people's time, what would they do with it? We have no real concept of what is possible.

Starting in the late 1880s, Taylor began a program of experimentation to discover the optimal way to cut steel. In the course of that research, which lasted more than twenty-five years, he and his colleagues performed more than twenty thousand individual experiments. What is remarkable about this project is that it had no academic backing, no government R&D budget. Its entire cost was paid by industry out of the immediate profits generated from the higher productivity the experiments enabled. This was only one experimental program to uncover the hidden productivity in just one kind of work. Other scientific management disciples spent years investigating bricklaying, farming, and even shoveling. They were obsessed with learning the truth and were not satisfied with the folk wisdom of craftspersons or the parables of experts.

Can any of us imagine a modern knowledge-work manager with the same level of interest in the methods his or her employees use?

How much of our current innovation work is guided by catchphrases that lack a scientific foundation?

## A New Research Program

What comparable research programs could we be engaged in to discover how to work more effectively?

For one thing, we have very little understanding of what stimulates productivity under conditions of extreme uncertainty. Luckily, with cycle times falling everywhere, we have many opportunities to test new approaches. Thus, I propose that we create startup testing labs that could put all manner of product development methodologies to the test.

How might those tests be conducted? We could bring in small cross-functional teams, perhaps beginning with product and engineering, and have them work to solve problems by using different development methodologies. We could begin with problems with clear right answers, perhaps drawn from the many international programming competitions that have developed databases of well-defined problems with clear solutions. These competitions also provide a clear baseline of how long it should take for various problems to be solved so that we could establish clearly the individual problem-solving prowess of the experimental subjects.

Using this kind of setup for calibration, we could begin to vary the conditions of the experiments. The challenge will be to increase the level of uncertainty about what the right answer is while still being able to measure the quality of the outcome objectively. Perhaps we could use real-world customer problems and then have real consumers test the output of the teams' work. Or perhaps we could go so far as to build minimum viable products for solving the same set of problems over and over again to quantify which produces the best customer conversion rates.

We also could vary the all-important cycle time by choosing more or less complex development platforms and distribution channels to

test the impact of those factors on the true productivity of the teams.

Most of all, we need to develop clear methods for holding teams accountable for validated learning. I have proposed one method in this book: innovation accounting using a well-defined financial model and engine of growth. However, it is naive to assume that this is the best possible method. As it is adopted in more and more companies, undoubtedly new techniques will be suggested, and we need to be able to evaluate the new ideas as rigorously as possible.

All these questions raise the possibilities of public-private partnerships between research universities and the entrepreneurial communities they seek to foster. It also suggests that universities may be able to add value in more ways than by being simply financial investors or creators of startup incubators, as is the current trend. My prediction is that wherever this research is conducted will become an epicenter of new entrepreneurial practice, and universities conducting this research therefore may be able to achieve a much higher level of commercialization of their basic research activities.<sup>4</sup>

## THE LONG-TERM STOCK EXCHANGE

Beyond simple research, I believe our goal should be to change the entire ecosystem of entrepreneurship. Too much of our startup industry has devolved into a feeder system for giant media companies and investment banks. Part of the reason established companies struggle to invest consistently in innovation is intense pressure from public markets to hit short-term profitability and growth targets. Mostly, this is a consequence of the accounting methods we have developed for evaluating managers, which focus on the kinds of gross “vanity” metrics discussed in [Chapter 7](#). What is needed is a new kind of stock exchange, designed to trade in the stocks of companies that are organized to sustain long-term thinking. I propose that we create a Long-Term Stock Exchange (LTSE).

In addition to quarterly reports on profits and margins, companies on the LTSE would report using innovation accounting on their internal entrepreneurship efforts. Like Intuit, they would report on the revenue they were generating from products that did not exist a few years earlier. Executive compensation in LTSE companies would be tied to the company's long-term performance. Trading on the LTSE would have much higher transaction costs and fees to minimize day trading and massive price swings. In exchange, LTSE companies would be allowed to structure their corporate governance to facilitate greater freedom for management to pursue long-term investments. In addition to support for long-term thinking, the transparency of the LTSE will provide valuable data about how to nurture innovation in the real world. Something like the LTSE would accelerate the creation of the next generation of great companies, built from the ground up for continuous innovation.

## IN CONCLUSION

As a movement, the Lean Startup must avoid doctrines and rigid ideology. We must avoid the caricature that science means formula or a lack of humanity in work. In fact, science is one of humanity's most creative pursuits. I believe that applying it to entrepreneurship will unlock a vast storehouse of human potential.

What would an organization look like if all of its employees were armed with Lean Startup organizational superpowers?

For one thing, everyone would insist that assumptions be stated explicitly and tested rigorously not as a stalling tactic or a form of make-work but out of a genuine desire to discover the truth that underlies every project's vision.

We would not waste time on endless arguments between the defenders of quality and the cowboys of reckless advance; instead, we would recognize that speed and quality are allies in the pursuit of the customer's long-term benefit. We would race to test our vision but not to abandon it. We would look to eliminate waste not

to build quality castles in the sky but in the service of agility and breakthrough business results.

We would respond to failures and setbacks with honesty and learning, not with recriminations and blame. More than that, we would shun the impulse to slow down, increase batch size, and indulge in the curse of prevention. Instead, we would achieve speed by bypassing the excess work that does not lead to learning. We would dedicate ourselves to the creation of new institutions with a long-term mission to build sustainable value and change the world for the better.

Most of all, we would stop wasting people's time.

## JOIN THE MOVEMENT

In the past few years, the Lean Startup movement has gone global. The number of resources available for aspiring entrepreneurs is incredible. Here, I'll do my best to list just a few of the best events, books, and blogs for further reading and further practice. The rest is up to you. Reading is good, action is better.

The most important resources are local. Gone are the days where you had to be in Silicon Valley to find other entrepreneurs to share ideas and struggles with. However, being embedded in a startup ecosystem is still an important part of entrepreneurship. What's changed is that these ecosystems are springing up in more and more startup hubs around the world.

I maintain an official website for The Lean Startup at <http://theleanstartup.com>, where you can find additional resources, including case studies and links to further reading. You will also find links there to my blog, Startup Lessons Learned, as well as videos, slides, and audio from my past presentations.

### Lean Startup Meetups

Chances are there is a Lean Startup meetup group near you. As of this writing, there are over a hundred, with the largest in San Francisco, Boston, New York, Chicago, and Los Angeles. You can find a real-time map of groups here: <http://lean-startup.meetup.com/>. You can also find a list of cities where people are interested in starting a new group, and tools to set one up



yourself.

## The Lean Startup Wiki

Not every Lean Startup group uses [Meetup.com](#) to organize, and a comprehensive list of events and other resources is maintained by volunteers on the Lean Startup Wiki: <http://leanstartup.pbworks.com/>

## The Lean Startup Circle

The largest community of practice around the Lean Startup is happening online, right now, on the Lean Startup Circle mailing list. Founded by Rich Collins, the list has thousands of entrepreneurs sharing tips, resources, and stories every day. If you have a question about how Lean Startup might apply to your business or industry, it's a great place to start: <http://leanstartupcircle.com/>

## The Startup Lessons Learned Conference

For the past two years, I have run a conference called Startup Lessons Learned. More details are available here: <http://slconf.com>

## REQUIRED READING

Steve Blank's book *The Four Steps to the Epiphany* is the original book about customer development. When I was building IMVU, a dog-eared copy of this book followed me everywhere. It is an indispensable guide. You can get a copy here: <http://ericri.es/FourSteps> or read my review of it here: <http://www.startuplessonslearned.com/2008/11/what-is-customer-development.html>. Steve also maintains an active and excellent

blog at <http://steveblank.com/>

Brant Cooper and Patrick Vlaskovits have created a short but excellent book called The Entrepreneur's Guide to Customer Development, which provides a gentle introduction to the topic. You can buy it here: <http://custdev.com> or read my review here: <http://www.startuplessonslearned.com/2010/07/entrepreneurs-guide-to-customer.html>

When I first began blogging about entrepreneurship, it was not nearly as common an occupation as it is now. Very few bloggers were actively working on new ideas about entrepreneurship, and together we debated and refined these ideas online.

Dave McClure, founder of the venture firm 500 Startups, writes a blog at <http://500hats.typepad.com/>. 500 Startups has an excellent blog as well: <http://blog.500startups.com/>. Dave's "Startup Metrics for Pirates" presentation laid out a framework for thinking about and measuring online services that greatly influenced the concept of "engines of growth." You can see the original presentation here: <http://500hats.typepad.com/500blogs/2008/09/startup-metrics-2.html> as well as my original reaction here: <http://www.startuplessonslearned.com/2008/09/three-drivers-of-growth-for-your.html>

Sean Ellis writes the Startup Marketing Blog, which has been influential in my thinking about how to integrate marketing into startups: <http://startup-marketing.com/>

Andrew Chen's blog Futuristic Play is one of the best sources for thoughts on viral marketing, startup metrics, and design: <http://andrewchenblog.com/>

Babak Nivi writes the excellent blog Venture Hacks and was an early Lean Startup evangelist: <http://venturehacks.com/>. He's since gone on to create Angel List, which matches startups and investors around the world: <http://angel.co/>

Other fantastic Lean Startup blogs include:

- Ash Maurya has emerged as a leader in helping bootstrapped online businesses apply Lean Startup ideas. His blog is called Running Lean, and he also has released an eBook of the same name. Both can be found here: <http://www.runningleanhq.com/>
- Sean Murphy on early-stage software startups: <http://www.skmurphy.com/blog/>
- Brant Cooper's Market by Numbers: <http://market-by-numbers.com/>
- Patrick Vlaskovits on technology, customer development, and pricing: <http://vlaskovits.com/>
- The KISSmetrics Marketing Blog: <http://blog.kissmetrics.com/> and Hiten Shah's <http://hitenism.com>

## FURTHER READING

Clayton M. Christensen's The Innovator's Dilemma and The Innovator's Solution are classics. In addition, Christensen's more recent work is also extremely helpful for seeing the theory of disruptive innovation in practice, including The Innovator's Prescription (about disrupting health care) and Disrupting Class (about education).

<http://ericri.es/ClaytonChristensen>

Geoffrey A. Moore's early work is famous among all entrepreneurs, especially Crossing the Chasm and Inside the Tornado. But he has continued to refine his thinking, and I have found his latest work, Dealing with Darwin: How Great Companies Innovate at Every Phase of Their Evolution, especially useful.

<http://ericri.es/DealingWithDarwin>

The Principles of Product Development Flow: Second Generation Lean Product Development by Donald G. Reinertsen.

<http://ericri.es/pdfflow>

The Toyota Way by Jeffrey Liker.

<http://ericri.es/thetoyotaway>

Lean Thinking: Banish Waste and Create Wealth in Your Corporation, Revised and Updated by James P. Womack and Daniel T. Jones.

<http://ericri.es/LeanThinking>

The People's Tycoon: Henry Ford and the American Century by Steven Watts.

<http://ericri.es/ThePeoplesTycoon>

The One Best Way: Frederick Winslow Taylor and the Enigma of Efficiency by Robert Kanigel.

<http://ericri.es/OneBestWay>

The Principles of Scientific Management by Frederick Winslow Taylor.

<http://ericri.es/ScientificManagement>

Extreme Programming Explained: Embrace Change by Kent Beck and Cynthia Andres.

<http://ericri.es/EmbraceChange>

Toyota Production System: Beyond Large-Scale Production by Taiichi Ohno.

<http://ericri.es/TaiichiOhno>

The idea of the Build-Measure-Learn feedback loop owes a lot to ideas from maneuver warfare, especially John Boyd's OODA (Observe-Orient-Decide-Act) Loop. The most accessible introduction to Boyd's ideas is Certain to Win: The Strategy of John Boyd, Applied to Business by Chet Richards.

<http://ericri.es/CertainToWin>

Out of the Crisis by W. Edwards Deming.

<http://ericri.es/OutOfTheCrisis>

My Years with General Motors by Alfred Sloan.

<http://ericri.es/MyYears>

Billy, Alfred, and General Motors: The Story of Two Unique Men, a Legendary Company, and a Remarkable Time in American History by William Pelfrey.

<http://ericri.es/BillyAlfred>

The Practice of Management by Peter F. Drucker.

<http://ericri.es/PracticeOfManagement>

Getting to Plan B: Breaking Through to a Better Business Model by John Mullins and Randy Komisar.

<http://ericri.es/GettingToPlanB>