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?"[1]

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.[2]

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.[3] It recently was named one of *Fast Company*'s ten most innovative companies in finance.[4] 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.[5]

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).[6] 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.

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.[1]

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[2] 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.[3]

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.