

---

## LEARN

**A**s an entrepreneur, nothing plagued me more than the question of whether my company was making progress toward creating a successful business. As an engineer and later as a manager, I was accustomed to measuring progress by making sure our work proceeded according to plan, was high quality, and cost about what we had projected.

After many years as an entrepreneur, I started to worry about measuring progress in this way. What if we found ourselves building something that nobody wanted? In that case what did it matter if we did it on time and on budget? When I went home at the end of a day's work, the only things I knew for sure were that I had kept people busy and spent money that day. I hoped that my team's efforts took us closer to our goal. If we wound up taking a wrong turn, I'd have to take comfort in the fact that at least we'd learned something important.

Unfortunately, "learning" is the oldest excuse in the book for a failure of execution. It's what managers fall back on when they fail to achieve the results we promised. Entrepreneurs, under pressure to succeed, are wildly creative when it comes to demonstrating what we have learned. We can all tell a good story when our job, career, or reputation depends on it.

However, learning is cold comfort to employees who are following an entrepreneur into the unknown. It is cold comfort to the investors who allocate precious money, time, and energy to entrepreneurial teams. It is cold comfort to the organizations—large and small—that depend on entrepreneurial innovation to survive. You can't take learning to the bank; you can't spend it or invest it. You cannot give it to customers and cannot return it to limited

partners. Is it any wonder that learning has a bad name in entrepreneurial and managerial circles?

Yet if the fundamental goal of entrepreneurship is to engage in organization building under conditions of extreme uncertainty, its most vital function is learning. We must learn the truth about which elements of our strategy are working to realize our vision and which are just crazy. We must learn what customers really want, not what they say they want or what we think they should want. We must discover whether we are on a path that will lead to growing a sustainable business.

In the Lean Startup model, we are rehabilitating learning with a concept I call *validated learning*. Validated learning is not after-the-fact rationalization or a good story designed to hide failure. It is a rigorous method for demonstrating progress when one is embedded in the soil of extreme uncertainty in which startups grow. Validated learning is the process of demonstrating empirically that a team has discovered valuable truths about a startup's present and future business prospects. It is more concrete, more accurate, and faster than market forecasting or classical business planning. It is the principal antidote to the lethal problem of achieving failure: successfully executing a plan that leads nowhere.

## **VALIDATED LEARNING AT IMVU**

Let me illustrate this with an example from my career. Many audiences have heard me recount the story of IMVU's founding and the many mistakes we made in developing our first product. I'll now elaborate on one of those mistakes to illustrate validated learning clearly.

Those of us involved in the founding of IMVU aspired to be serious strategic thinkers. Each of us had participated in previous ventures that had failed, and we were loath to repeat that experience. Our main concerns in the early days dealt with the following questions: What should we build and for whom? What market could we enter and dominate? How could we build durable value that would not be subject to erosion by competition?<sup>1</sup>

## Brilliant Strategy

We decided to enter the instant messaging (IM) market. In 2004, that market had hundreds of millions of consumers actively participating worldwide. However, the majority of the customers who were using IM products were not paying for the privilege. Instead, large media and portal companies such as AOL, Microsoft, and Yahoo! operated their IM networks as a loss leader for other services while making modest amounts of money through advertising.

IM is an example of a market that involves strong *network effects*. Like most communication networks, IM is thought to follow Metcalfe's law: the value of a network as a whole is proportional to the square of the number of participants. In other words, the more people in the network, the more valuable the network. This makes intuitive sense: the value to each participant is driven primarily by how many other people he or she can communicate with. Imagine a world in which you own the only telephone; it would have no value. Only when other people also have a telephone does it become valuable.

In 2004, the IM market was locked up by a handful of incumbents. The top three networks controlled more than 80 percent of the overall usage and were in the process of consolidating their gains in market share at the expense of a number of smaller players.<sup>2</sup> The common wisdom was that it was more or less impossible to bring a new IM network to market without spending an extraordinary amount of money on marketing.

The reason for that wisdom is simple. Because of the power of network effects, IM products have high switching costs. To switch from one network to another, customers would have to convince their friends and colleagues to switch with them. This extra work for customers creates a barrier to entry in the IM market: with all consumers locked in to an incumbent's product, there are no customers left with whom to establish a beachhead.

At IMVU we settled on a strategy of building a product that would combine the large mass appeal of traditional IM with the high

revenue per customer of three-dimensional (3D) video games and virtual worlds. Because of the near impossibility of bringing a new IM network to market, we decided to build an IM add-on product that would interoperate with the existing networks. Thus, customers would be able to adopt the IMVU virtual goods and avatar communication technology without having to switch IM providers, learn a new user interface, and—most important—bring their friends with them.

In fact, we thought this last point was essential. For the add-on product to be useful, customers would *have* to use it with their existing friends. Every communication would come embedded with an invitation to join IMVU. Our product would be inherently viral, spreading throughout the existing IM networks like an epidemic. To achieve that viral growth, it was important that our add-on product support as many of the existing IM networks as possible and work on all kinds of computers.

## **Six Months to Launch**

With this strategy in place, my cofounders and I began a period of intense work. As the chief technology officer, it was my responsibility, among other things, to write the software that would support IM interoperability across networks. My cofounders and I worked for months, putting in crazy hours struggling to get our first product released. We gave ourselves a hard deadline of six months—180 days—to launch the product and attract our first paying customers. It was a grueling schedule, but we were determined to launch on time.

The add-on product was so large and complex and had so many moving parts that we had to cut a lot of corners to get it done on time. I won't mince words: the first version was terrible. We spent endless hours arguing about which bugs to fix and which we could live with, which features to cut and which to try to cram in. It was a wonderful and terrifying time: we were full of hope about the possibilities for success and full of fear about the consequences of shipping a bad product.

Personally, I was worried that the low quality of the product would tarnish my reputation as an engineer. People would think I didn't know how to build a quality product. All of us feared tarnishing the IMVU brand; after all, we were charging people money for a product that didn't work very well. We all envisioned the damning newspaper headlines: "Inept Entrepreneurs Build Dreadful Product."

As launch day approached, our fears escalated. In our situation, many entrepreneurial teams give in to fear and postpone the launch date. Although I understand this impulse, I am glad we persevered, since delay prevents many startups from getting the feedback they need. Our previous failures made us more afraid of another, even worse, outcome than shipping a bad product: building something that nobody wants. And so, teeth clenched and apologies at the ready, we released our product to the public.

## **Launch**

And then—nothing happened! It turned out that our fears were unfounded, because nobody even tried our product. At first I was relieved because at least nobody was finding out how bad the product was, but soon that gave way to serious frustration. After all the hours we had spent arguing about which features to include and which bugs to fix, our value proposition was so far off that customers weren't getting far enough into the experience to find out how bad our design choices were. Customers wouldn't even download our product.

Over the ensuing weeks and months, we labored to make the product better. We brought in a steady flow of customers through our online registration and download process. We treated each day's customers as a brand-new report card to let us know how we were doing. We eventually learned how to change the product's positioning so that customers at least would download it. We were making improvements to the underlying product continuously, shipping bug fixes and new changes daily. However, despite our

best efforts, we were able to persuade only a pathetically small number of people to buy the product.

In retrospect, one good decision we made was to set clear revenue targets for those early days. In the first month we intended to make \$300 in total revenue, and we did—barely. Many friends and family members were asked (okay, begged). Each month our small revenue targets increased, first to \$350 and then to \$400. As they rose, our struggles increased. We soon ran out of friends and family; our frustration escalated. We were making the product better every day, yet our customers' behavior remained unchanged: they still wouldn't use it.

Our failure to move the numbers prodded us to accelerate our efforts to bring customers into our office for in-person interviews and usability tests. The quantitative targets created the motivation to engage in qualitative inquiry and guided us in the questions we asked; this is a pattern we'll see throughout this book.

I wish I could say that I was the one to realize our mistake and suggest the solution, but in truth, I was the last to admit the problem. In short, our entire strategic analysis of the market was utterly wrong. We figured this out empirically, through experimentation, rather than through focus groups or market research. Customers could not tell us what they wanted; most, after all, had never heard of 3D avatars. Instead, they revealed the truth through their action or inaction as we struggled to make the product better.

## **Talking to Customers**

Out of desperation, we decided to talk to some potential customers. We brought them into our office, and said, "Try this new product; it's IMVU." If the person was a teenager, a heavy user of IM, or a tech early adopter, he or she would engage with us. In contrast, if it was a mainstream person, the response was, "Right. So exactly what would you like me to do?" We'd get nowhere with the mainstream group; they thought IMVU was too weird.

Imagine a seventeen-year-old girl sitting down with us to look at this product. She chooses her avatar and says, “Oh, this is really fun.” She’s customizing the avatar, deciding how she’s going to look. Then we say, “All right, it’s time to download the instant messaging add-on,” and she responds, “What’s that?”

“Well, it’s this thing that interoperates with the instant messaging client.” She’s looking at us and thinking, “I’ve never heard of that, my friends have never heard of that, why do you want me to do that?” It required a lot of explanation; an instant messaging add-on was not a product category that existed in her mind.

But since she was in the room with us, we were able to talk her into doing it. She downloads the product, and then we say, “Okay, invite one of your friends to chat.” And she says, “No way!” We say, “Why not?” And she says, “Well, I don’t know if this thing is cool yet. You want me to risk inviting one of my friends? What are they going to think of me? If it sucks, they’re going to think I suck, right?” And we say, “No, no, it’s going to be so much fun once you get the person in there; it’s a *social* product.” She looks at us, her face filled with doubt; you can see that this is a deal breaker. Of course, the first time I had that experience, I said, “It’s all right, it’s just this one person, send her away and get me a new one.” Then the second customer comes in and says the same thing. Then the third customer comes in, and it’s the same thing. You start to see patterns, and no matter how stubborn you are, there’s obviously something wrong.

Customers kept saying, “I want to use it by myself. I want to try it out first to see if it’s really cool before I invite a friend.” Our team was from the video game industry, so we understood what that meant: single-player mode. So we built a single-player version. We’d bring new customers into our office. They’d customize the avatar and download the product like before. Then they would go into single-player mode, and we’d say, “Play with your avatar and dress it up; check out the cool moves it can make.” Followed by, “Okay, you did that by yourself; now it’s time to invite one of your friends.” You can see what’s coming. They’d say, “No way! This isn’t cool.” And we’d say, “Well, we *told* you it wasn’t going to be cool! What is

the point of a single-player experience for a social product?” See, we thought we should get a gold star just for listening to our customers. Except our customers still didn’t like the product. They would look at us and say, “Listen, old man, you don’t understand. What is the deal with this crazy business of inviting friends before I know if it’s cool?” It was a total deal breaker.

Out of further desperation, we introduced a feature called ChatNow that allows you to push a button and be randomly matched with somebody else anywhere in the world. The only thing you have in common is that you both pushed the button at the same time. All of a sudden, in our customer service tests, people were saying, “Oh, this is fun!”

So we’d bring them in, they’d use ChatNow, and maybe they would meet somebody they thought was cool. They’d say, “Hey, that guy was neat; I want to add him to my buddy list. Where’s my buddy list?” And we’d say, “Oh, no, you don’t want a new buddy list; you want to use your regular AOL buddy list.” Remember, this was how we planned to harness the interoperability that would lead to network effects and viral growth. Picture the customer looking at us, asking, “What do you want me to do exactly?” And we’d say, “Well, just give the stranger your AIM screen name so you can put him on your buddy list.” You could see their eyes go wide, and they’d say, “Are you kidding me? A stranger on my AIM buddy list?” To which we’d respond, “Yes; otherwise you’d have to download a whole new IM client with a new buddy list.” And they’d say, “Do you have any idea how many IM clients I already run?”

“No. One or two, maybe?” That’s how many clients each of us in the office used. To which the teenager would say, “Duh! I run eight.” We had no idea how many instant messaging clients there were in the world.

We had the incorrect preconception that it’s a challenge to learn new software and it’s tricky to move your friends over to a new buddy list. Our customers revealed that this was nonsense. We wanted to draw diagrams on the whiteboard that showed why our strategy was brilliant, but our customers didn’t understand concepts like network effects and switching costs. If we tried to explain why

they should behave the way we predicted, they'd just shake their heads at us, bewildered.

We had a mental model for how people used software that was years out of date, and so eventually, painfully, after dozens of meetings like that, it started to dawn on us that the IM add-on concept was fundamentally flawed.<sup>3</sup>

Our customers did not want an IM add-on; they wanted a stand-alone IM network. They did not consider having to learn how to use a new IM program a barrier; on the contrary, our early adopters used many different IM programs simultaneously. Our customers were not intimidated by the idea of having to take their friends with them to a new IM network; it turned out that they enjoyed that challenge. Even more surprising, our assumption that customers would want to use avatar-based IM primarily with their existing friends was also wrong. They wanted to make new friends, an activity that 3D avatars are particularly well suited to facilitating. Bit by bit, customers tore apart our seemingly brilliant initial strategy.

## **Throwing My Work Away**

Perhaps you can sympathize with our situation and forgive my obstinacy. After all, it was my work over the prior months that needed to be thrown away. I had slaved over the software that was required to make our IM program interoperate with other networks, which was at the heart of our original strategy. When it came time to pivot and abandon that original strategy, almost all of my work—thousands of lines of code—was thrown out. I felt betrayed. I was a devotee of the latest in software development methods (known collectively as agile development), which promised to help drive waste out of product development. However, despite that, I had committed the biggest waste of all: building a product that our customers refused to use. That was *really* depressing.

I wondered: in light of the fact that my work turned out to be a waste of time and energy, would the company have been just as well off if I had spent the last six months on a beach sipping

umbrella drinks? Had I really been needed? Would it have been better if I had not done any work at all?

There is, as I mentioned at the beginning of this chapter, always one last refuge for people aching to justify their own failure. I consoled myself that if we hadn't built this first product—mistakes and all—we never would have learned these important insights about customers. We never would have learned that our strategy was flawed. There is truth in this excuse: what we learned during those critical early months set IMVU on a path that would lead to our eventual breakout success.

For a time, this “learning” consolation made me feel better, but my relief was short-lived. Here's the question that bothered me most of all: if the goal of those months was to learn these important insights about customers, why did it take so long? How much of our effort contributed to the essential lessons we needed to learn? Could we have learned those lessons earlier if I hadn't been so focused on making the product “better” by adding features and fixing bugs?

## **VALUE VS. WASTE**

In other words, which of our efforts are value-creating and which are wasteful? This question is at the heart of the lean manufacturing revolution; it is the first question any lean manufacturing adherent is trained to ask. Learning to see waste and then systematically eliminate it has allowed lean companies such as Toyota to dominate entire industries. In the world of software, the agile development methodologies I had practiced until that time had their origins in lean thinking. They were designed to eliminate waste too.

Yet those methods had led me down a road in which the majority of my team's efforts were wasted. Why?

The answer came to me slowly over the subsequent years. Lean thinking defines value as providing benefit to the customer; anything else is waste. In a manufacturing business, customers don't care how the product is assembled, only that it works correctly. But in a startup, who the customer is and what the customer might find valuable are unknown, part of the very uncertainty that is an

essential part of the definition of a startup. I realized that as a startup, we needed a new definition of value. The real progress we had made at IMVU was what we had learned over those first months about what creates value for customers.

Anything we had done during those months that did not contribute to our learning was a form of waste. Would it have been possible to learn the same things with less effort? Clearly, the answer is yes.

For one thing, think of all the debate and prioritization of effort that went into features that customers would never discover. If we had shipped sooner, we could have avoided that waste. Also consider all the waste caused by our incorrect strategic assumptions. I had built interoperability for more than a dozen different IM clients and networks. Was this really necessary to test our assumptions? Could we have gotten the same feedback from our customers with half as many networks? With only three? With only one? Since the customers of all IM networks found our product equally unattractive, the level of learning would have been the same, but our effort would have been dramatically less.

Here's the thought that kept me up nights: did we have to support any networks at all? Is it possible that we could have discovered how flawed our assumptions were without building anything? For example, what if we simply had offered customers the opportunity to download the product from us solely on the basis of its proposed features before building anything? Remember, almost no customers were willing to use our original product, so we wouldn't have had to do much apologizing when we failed to deliver. (Note that this is different from asking customers what they want. Most of the time customers don't know what they want in advance.) We could have conducted an experiment, offering customers the chance to try something and then measuring their behavior.

Such thought experiments were extremely disturbing to me because they undermined my job description. As the head of product development, I thought my job was to ensure the timely delivery of high-quality products and features. But if many of those

features were a waste of time, what should I be doing instead? How could we avoid this waste?

I've come to believe that learning is the essential unit of progress for startups. The effort that is not absolutely necessary for learning what customers want can be eliminated. I call this *validated learning* because it is always demonstrated by positive improvements in the startup's core metrics. As we've seen, it's easy to kid yourself about what you think customers want. It's also easy to learn things that are completely irrelevant. Thus, validated learning is backed up by empirical data collected from real customers.

## **WHERE DO YOU FIND VALIDATION?**

As I can attest, anybody who fails in a startup can claim that he or she has learned a lot from the experience. They can tell a compelling story. In fact, in the story of IMVU so far, you might have noticed something missing. Despite my claims that we learned a lot in those early months, lessons that led to our eventual success, I haven't offered any evidence to back that up. In hindsight, it's easy to make such claims and sound credible (and you'll see some evidence later in the book), but imagine us in IMVU's early months trying to convince investors, employees, family members, and most of all ourselves that we had not squandered our time and resources. What evidence did we have?

Certainly our stories of failure were entertaining, and we had fascinating theories about what we had done wrong and what we needed to do to create a more successful product. However, the proof did not come until we put those theories into practice and built subsequent versions of the product that showed superior results with actual customers.

The next few months are where the true story of IMVU begins, not with our brilliant assumptions and strategies and whiteboard gamesmanship but with the hard work of discovering what customers really wanted and adjusting our product and strategy to meet those desires. We adopted the view that our job was to find a synthesis between our vision and what customers would accept; it

wasn't to capitulate to what customers thought they wanted or to tell customers what they ought to want.

As we came to understand our customers better, we were able to improve our products. As we did that, the fundamental metrics of our business changed. In the early days, despite our efforts to improve the product, our metrics were stubbornly flat. We treated each day's customers as a new report card. We'd pay attention to the percentage of new customers who exhibited product behaviors such as downloading and buying our product. Each day, roughly the same number of customers would buy the product, and that number was pretty close to zero despite the many improvements.

However, once we pivoted away from the original strategy, things started to change. Aligned with a superior strategy, our product development efforts became magically more productive—not because we were working harder but because we were working smarter, aligned with our customers' real needs. Positive changes in metrics became the quantitative validation that our learning was real. This was critically important because we could show our stakeholders—employees, investors, and ourselves—that we were making genuine progress, not deluding ourselves. It is also the right way to think about productivity in a startup: not in terms of how much stuff we are building but in terms of how much validated learning we're getting for our efforts.<sup>4</sup>

For example, in one early experiment, we changed our entire website, home page, and product registration flow to replace “avatar chat” with “3D instant messaging.” New customers were split automatically between these two versions of the site; half saw one, and half saw the other. We were able to measure the difference in behavior between the two groups. Not only were the people in the experimental group more likely to sign up for the product, they were more likely to become long-term paying customers.

We had plenty of failed experiments too. During one period in which we believed that customers weren't using the product because they didn't understand its many benefits, we went so far as to pay customer service agents to act as virtual tour guides for new

customers. Unfortunately, customers who got that VIP treatment were no more likely to become active or paying customers.

Even after ditching the IM add-on strategy, it still took months to understand *why* it hadn't worked. After our pivot and many failed experiments, we finally figured out this insight: customers wanted to use IMVU to make *new* friends online. Our customers intuitively grasped something that we were slow to realize. All the existing social products online were centered on customers' real-life identity. IMVU's avatar technology, however, was uniquely well suited to help people get to know each other online without compromising safety or opening themselves up to identity theft. Once we formed this hypothesis, our experiments became much more likely to produce positive results. Whenever we would change the product to make it easier for people to find and keep new friends, we discovered that customers were more likely to engage. This is true startup productivity: systematically figuring out the right things to build.

These were just a few experiments among hundreds that we ran week in and week out as we started to learn which customers would use the product and why. Each bit of knowledge we gathered suggested new experiments to run, which moved our metrics closer and closer to our goal.

## **THE AUDACITY OF ZERO**

Despite IMVU's early success, our gross numbers were still pretty small. Unfortunately, because of the traditional way businesses are evaluated, this is a dangerous situation. The irony is that it is often easier to raise money or acquire other resources when you have zero revenue, zero customers, and zero traction than when you have a small amount. Zero invites imagination, but small numbers invite questions about whether large numbers will ever materialize. Everyone knows (or thinks he or she knows) stories of products that achieved breakthrough success overnight. As long as nothing has been released and no data have been collected, it is still possible to

imagine overnight success in the future. Small numbers pour cold water on that hope.

This phenomenon creates a brutal incentive: postpone getting any data until you are certain of success. Of course, as we'll see, such delays have the unfortunate effect of increasing the amount of wasted work, decreasing essential feedback, and dramatically increasing the risk that a startup will build something nobody wants.

However, releasing a product and hoping for the best is not a good plan either, because this incentive is real. When we launched IMVU, we were ignorant of this problem. Our earliest investors and advisers thought it was quaint that we had a \$300-per-month revenue plan at first. But after several months with our revenue hovering around \$500 per month, some began to lose faith, as did some of our advisers, employees, and even spouses. In fact, at one point, some investors were seriously recommending that we pull the product out of the market and return to stealth mode. Fortunately, as we pivoted and experimented, incorporating what we learned into our product development and marketing efforts, our numbers started to improve.

But not by much! On the one hand, we were lucky to see a growth pattern that started to look like the famous hockey stick graph. On the other hand, the graph went up only to a few thousand dollars per month. These early graphs, although promising, were not by themselves sufficient to combat the loss of faith caused by our early failure, and we lacked the language of validated learning to provide an alternative concept to rally around. We were quite fortunate that some of our early investors understood its importance and were willing to look beyond our small gross numbers to see the real progress we were making. (You'll see the exact same graphs they did in [Chapter 7](#).)

Thus, we can mitigate the waste that happens because of the audacity of zero with validated learning. What we needed to demonstrate was that our product development efforts were leading us toward massive success without giving in to the temptation to fall back on vanity metrics and “success theater”—the work we do to

make ourselves look successful. We could have tried marketing gimmicks, bought a Super Bowl ad, or tried flamboyant public relations (PR) as a way of juicing our gross numbers. That would have given investors the illusion of traction, but only for a short time. Eventually, the fundamentals of the business would win out and the PR bump would pass. Because we would have squandered precious resources on theatrics instead of progress, we would have been in real trouble.

Sixty million avatars later, IMVU is still going strong. Its legacy is not just a great product, an amazing team, and promising financial results but a whole new way of measuring the progress of startups.

## **LESSONS BEYOND IMVU**

I have had many opportunities to teach the IMVU story as a business case ever since Stanford's Graduate School of Business wrote an official study about IMVU's early years.<sup>5</sup> The case is now part of the entrepreneurship curriculum at several business schools, including Harvard Business School, where I serve as an entrepreneur in residence. I've also told these stories at countless workshops, lectures, and conferences.

Every time I teach the IMVU story, students have an overwhelming temptation to focus on the tactics it illustrates: launching a low-quality early prototype, charging customers from day one, and using low-volume revenue targets as a way to drive accountability. These are useful techniques, but they are not the moral of the story. There are too many exceptions. Not every kind of customer will accept a low-quality prototype, for example. If the students are more skeptical, they may argue that the techniques do not apply to their industry or situation, but work only because IMVU is a software company, a consumer Internet business, or a non-mission-critical application.

None of these takeaways is especially useful. The Lean Startup is not a collection of individual tactics. It is a principled approach to new product development. The only way to make sense of its recommendations is to understand the underlying principles that

make them work. As we'll see in later chapters, the Lean Startup model has been applied to a wide variety of businesses and industries: manufacturing, clean tech, restaurants, and even laundry. The tactics from the IMVU story may or may not make sense in your particular business.

Instead, the way forward is to learn to see every startup in any industry as a grand experiment. The question is not “Can this product be built?” In the modern economy, almost any product that can be imagined can be built. The more pertinent questions are “Should this product be built?” and “Can we build a sustainable business around this set of products and services?” To answer those questions, we need a method for systematically breaking down a business plan into its component parts and testing each part empirically.

In other words, we need the scientific method. In the Lean Startup model, every product, every feature, every marketing campaign—everything a startup does—is understood to be an experiment designed to achieve validated learning. This experimental approach works across industries and sectors, as we'll see in [Chapter 4](#).

---

## MEASURE

**A**t the beginning, a startup is little more than a model on a piece of paper. The financials in the business plan include projections of how many customers the company expects to attract, how much it will spend, and how much revenue and profit that will lead to. It's an ideal that's usually far from where the startup is in its early days.

A startup's job is to (1) rigorously measure where it is right now, confronting the hard truths that assessment reveals, and then (2) devise experiments to learn how to move the real numbers closer to the ideal reflected in the business plan.

Most products—even the ones that fail—do not have zero traction. Most products have some customers, some growth, and some positive results. One of the most dangerous outcomes for a startup is to bumble along in the land of the living dead. Employees and entrepreneurs tend to be optimistic by nature. We want to keep believing in our ideas even when the writing is on the wall. This is why the myth of perseverance is so dangerous. We all know stories of epic entrepreneurs who managed to pull out a victory when things seemed incredibly bleak. Unfortunately, we don't hear stories about the countless nameless others who persevered too long, leading their companies to failure.

### WHY SOMETHING AS SEEMINGLY DULL AS ACCOUNTING WILL CHANGE YOUR LIFE

People are accustomed to thinking of accounting as dry and boring, a necessary evil used primarily to prepare financial reports and survive audits, but that is because accounting is something that has

become taken for granted. Historically, under the leadership of people such as Alfred Sloan at General Motors, accounting became an essential part of the method of exerting centralized control over far-flung divisions. Accounting allowed GM to set clear milestones for each of its divisions and then hold each manager accountable for his or her division's success in reaching those goals. All modern corporations use some variation of that approach. Accounting is the key to their success.

Unfortunately, standard accounting is not helpful in evaluating entrepreneurs. Startups are too unpredictable for forecasts and milestones to be accurate.

I recently met with a phenomenal startup team. They are well financed, have significant customer traction, and are growing rapidly. Their product is a leader in an emerging category of enterprise software that uses consumer marketing techniques to sell into large companies. For example, they rely on employee-to-employee viral adoption rather than a traditional sales process, which might target the chief information officer or the head of information technology (IT). As a result, they have the opportunity to use cutting-edge experimental techniques as they constantly revise their product. During the meeting, I asked the team a simple question that I make a habit of asking startups whenever we meet: are you making your product better? They always say yes. Then I ask: how do you know? I invariably get this answer: well, we are in engineering and we made a number of changes last month, and our customers seem to like them, and our overall numbers are higher this month. We must be on the right track.

This is the kind of storytelling that takes place at most startup board meetings. Most milestones are built the same way: hit a certain product milestone, maybe talk to a few customers, and see if the numbers go up. Unfortunately, this is not a good indicator of whether a startup is making progress. How do we know that the changes we've made are related to the results we're seeing? More important, how do we know that we are drawing the right lessons from those changes?

To answer these kinds of questions, startups have a strong need for a new kind of accounting geared specifically to disruptive innovation. That's what innovation accounting is.

## **An Accountability Framework That Works Across Industries**

Innovation accounting enables startups to prove objectively that they are learning how to grow a sustainable business. Innovation accounting begins by turning the leap-of-faith assumptions discussed in [Chapter 5](#) into a quantitative financial model. Every business plan has some kind of model associated with it, even if it's written on the back of a napkin. That model provides assumptions about what the business will look like at a successful point in the future.

For example, the business plan for an established manufacturing company would show it growing in proportion to its sales volume. As the profits from the sales of goods are reinvested in marketing and promotions, the company gains new customers. The rate of growth depends primarily on three things: the profitability of each customer, the cost of acquiring new customers, and the repeat purchase rate of existing customers. The higher these values are, the faster the company will grow and the more profitable it will be. These are the drivers of the company's growth model.

By contrast, a marketplace company that matches buyers and sellers such as eBay will have a different growth model. Its success depends primarily on the network effects that make it the premier destination for both buyers and sellers to transact business. Sellers want the marketplace with the highest number of potential customers. Buyers want the marketplace with the most competition among sellers, which leads to the greatest availability of products and the lowest prices. (In economics, this sometimes is called supply-side increasing returns and demand-side increasing returns.) For this kind of startup, the important thing to measure is that the network effects are working, as evidenced by the high retention rate of new buyers and sellers. If people stick with the product with very little attrition, the marketplace will grow no matter how the company acquires new customers. The growth curve will look like a

compounding interest table, with the rate of growth depending on the “interest rate” of new customers coming to the product.

Though these two businesses have very different drivers of growth, we can still use a common framework to hold their leaders accountable. This framework supports accountability even when the model changes.

## **HOW INNOVATION ACCOUNTING WORKS—THREE LEARNING MILESTONES**

Innovation accounting works in three steps: first, use a minimum viable product to establish real data on where the company is right now. Without a clear-eyed picture of your current status—no matter how far from the goal you may be—you cannot begin to track your progress.

Second, startups must attempt to tune the engine from the baseline toward the ideal. This may take many attempts. After the startup has made all the micro changes and product optimizations it can to move its baseline toward the ideal, the company reaches a decision point. That is the third step: pivot or persevere.

If the company is making good progress toward the ideal, that means it’s learning appropriately and using that learning effectively, in which case it makes sense to continue. If not, the management team eventually must conclude that its current product strategy is flawed and needs a serious change. When a company pivots, it starts the process all over again, reestablishing a new baseline and then tuning the engine from there. The sign of a successful pivot is that these engine-tuning activities are more productive after the pivot than before.

### **Establish the Baseline**

For example, a startup might create a complete prototype of its product and offer to sell it to real customers through its main marketing channel. This single MVP would test most of the startup’s assumptions and establish baseline metrics for each assumption

simultaneously. Alternatively, a startup might prefer to build separate MVPs that are aimed at getting feedback on one assumption at a time. Before building the prototype, the company might perform a smoke test with its marketing materials. This is an old direct marketing technique in which customers are given the opportunity to preorder a product that has not yet been built. A smoke test measures only one thing: whether customers are interested in trying a product. By itself, this is insufficient to validate an entire growth model. Nonetheless, it can be very useful to get feedback on this assumption before committing more money and other resources to the product.

These MVPs provide the first example of a *learning milestone*. An MVP allows a startup to fill in real baseline data in its growth model—conversion rates, sign-up and trial rates, customer lifetime value, and so on—and this is valuable as the foundation for learning about customers and their reactions to a product even if that foundation begins with extremely bad news.

When one is choosing among the many assumptions in a business plan, it makes sense to test the riskiest assumptions first. If you can't find a way to mitigate these risks toward the ideal that is required for a sustainable business, there is no point in testing the others. For example, a media business that is selling advertising has two basic assumptions that take the form of questions: Can it capture the attention of a defined customer segment on an ongoing basis? and can it sell that attention to advertisers? In a business in which the advertising rates for a particular customer segment are well known, the far riskier assumption is the ability to capture attention. Therefore, the first experiments should involve content production rather than advertising sales. Perhaps the company will produce a pilot episode or issue to see how customers engage.

## **Tuning the Engine**

Once the baseline has been established, the startup can work toward the second learning milestone: tuning the engine. Every product development, marketing, or other initiative that a startup

undertakes should be targeted at improving one of the drivers of its growth model. For example, a company might spend time improving the design of its product to make it easier for new customers to use. This presupposes that the *activation rate* of new customers is a driver of growth and that its baseline is lower than the company would like. To demonstrate validated learning, the design changes must improve the activation rate of new customers. If they do not, the new design should be judged a failure. This is an important rule: a good design is one that changes customer behavior for the better.

Compare two startups. The first company sets out with a clear baseline metric, a hypothesis about what will improve that metric, and a set of experiments designed to test that hypothesis. The second team sits around debating what would improve the product, implements several of those changes at once, and celebrates if there is any positive increase in any of the numbers. Which startup is more likely to be doing effective work and achieving lasting results?

## **Pivot or Persevere**

Over time, a team that is learning its way toward a sustainable business will see the numbers in its model rise from the horrible baseline established by the MVP and converge to something like the ideal one established in the business plan. A startup that fails to do so will see that ideal recede ever farther into the distance. When this is done right, even the most powerful reality distortion field won't be able to cover up this simple fact: if we're not moving the drivers of our business model, we're not making progress. That becomes a sure sign that it's time to pivot.

## **INNOVATION ACCOUNTING AT IMVU**

Here's what innovation accounting looked like for us in the early days of IMVU. Our minimum viable product had many defects and, when we first released it, extremely low sales. We naturally assumed that the lack of sales was related to the low quality of the product,

so week after week we worked on improving the quality of the product, trusting that our efforts were worthwhile. At the end of each month, we would have a board meeting at which we would present the results. The night before the board meeting, we'd run our standard analytics, measuring conversion rates, customer counts, and revenue to show what a good job we had done. For several meetings in a row, this caused a last-minute panic because the quality improvements were not yielding any change in customer behavior. This led to some frustrating board meetings at which we could show great product “progress” but not much in the way of business results. After a while, rather than leave it to the last minute, we began to track our metrics more frequently, tightening the feedback loop with product development. This was even more depressing. Week in, week out, our product changes were having no effect.

### **Improving a Product on Five Dollars a Day**

We tracked the “funnel metrics” behaviors that were critical to our engine of growth: customer registration, the download of our application, trial, repeat usage, and purchase. To have enough data to learn, we needed just enough customers using our product to get real numbers for each behavior. We allocated a budget of five dollars per day: enough to buy clicks on the then-new Google AdWords system. In those days, the minimum you could bid for a click was 5 cents, but there was no overall minimum to your spending. Thus, we could afford to open an account and get started even though we had very little money.<sup>1</sup>

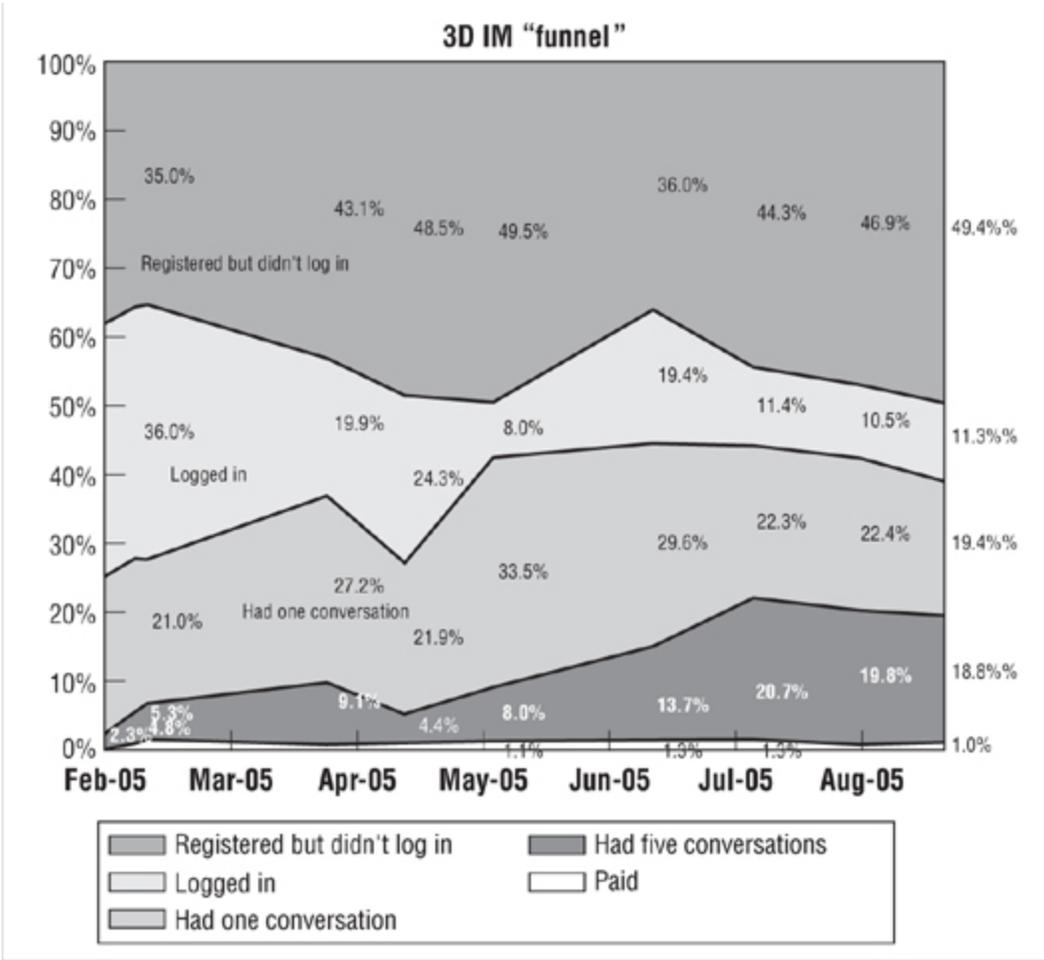
Five dollars bought us a hundred clicks—every day. From a marketing point of view this was not very significant, but for learning it was priceless. Every single day we were able to measure our product's performance with a brand new set of customers. Also, each time we revised the product, we got a brand new report card on how we were doing the very next day.

For example, one day we would debut a new marketing message aimed at first-time customers. The next day we might change the

way new customers were initiated into the product. Other days, we would add new features, fix bugs, roll out a new visual design, or try a new layout for our website. Every time, we told ourselves we were making the product better, but that subjective confidence was put to the acid test of real numbers.

Day in and day out we were performing random trials. Each day was a new experiment. Each day's customers were independent of those of the day before. Most important, even though our gross numbers were growing, it became clear that our funnel metrics were not changing.

Here is a graph from one of IMVU's early board meetings:



This graph represents approximately seven months of work. Over that period, we were making constant improvements to the IMVU product, releasing new features on a daily basis. We were

conducting a lot of in-person customer interviews, and our product development team was working extremely hard.

## **Cohort Analysis**

To read the graph, you need to understand something called *cohort analysis*. This is one of the most important tools of startup analytics. Although it sounds complex, it is based on a simple premise. Instead of looking at cumulative totals or gross numbers such as total revenue and total number of customers, one looks at the performance of each group of customers that comes into contact with the product independently. Each group is called a cohort. The graph shows the conversion rates to IMVU of new customers who joined in each indicated month. Each conversion rate shows the percentage of customer who registered in that month who subsequently went on to take the indicated action. Thus, among all the customers who joined IMVU in February 2005, about 60 percent of them logged in to our product at least one time.

Managers with an enterprise sales background will recognize this funnel analysis as the traditional sales funnel that is used to manage prospects on their way to becoming customers. Lean Startups use it in product development, too. This technique is useful in many types of business, because every company depends for its survival on sequences of customer behavior called flows. Customer flows govern the interaction of customers with a company's products. They allow us to understand a business quantitatively and have much more predictive power than do traditional gross metrics.

If you look closely, you'll see that the graph shows some clear trends. Some product improvements are helping—a little. The percentage of new customers who go on to use the product at least five times has grown from less than 5 percent to almost 20 percent. Yet despite this fourfold increase, the percentage of new customers who pay money for IMVU is stuck at around 1 percent. Think about that for a moment. After months and months of work, thousands of individual improvements, focus groups, design sessions, and usability tests, the percentage of new customers who subsequently

pay money is exactly the same as it was at the onset even though many more customers are getting a chance to try the product.

Thanks to the power of cohort analysis, we could not blame this failure on the legacy of previous customers who were resistant to change, external market conditions, or any other excuse. Each cohort represented an independent report card, and try as we might, we were getting straight C's. This helped us realize we had a problem.

I was in charge of the product development team, small though it was in those days, and shared with my cofounders the sense that the problem had to be with my team's efforts. I worked harder, tried to focus on higher- and higher-quality features, and lost a lot of sleep. Our frustration grew. When I could think of nothing else to do, I was finally ready to turn to the last resort: talking to customers. Armed with our failure to make progress tuning our engine of growth, I was ready to ask the right questions.

Before this failure, in the company's earliest days, it was easy to talk to potential customers and come away convinced we were on the right track. In fact, when we would invite customers into the office for in-person interviews and usability tests, it was easy to dismiss negative feedback. If they didn't want to use the product, I assumed they were not in our target market. "Fire that customer," I'd say to the person responsible for recruiting for our tests. "Find me someone in our target demographic." If the next customer was more positive, I would take it as confirmation that I was right in my targeting. If not, I'd fire another customer and try again.

By contrast, once I had data in hand, my interactions with customers changed. Suddenly I had urgent questions that needed answering: Why aren't customers responding to our product "improvements"? Why isn't our hard work paying off? For example, we kept making it easier and easier for customers to use IMVU with their existing friends. Unfortunately, customers didn't want to engage in that behavior. Making it easier to use was totally beside the point. Once we knew what to look for, genuine understanding came much faster. As was described in [Chapter 3](#), this eventually led to a critically important pivot: away from an IM add-on used with

existing friends and toward a stand-alone network one can use to make new friends. Suddenly, our worries about productivity vanished. Once our efforts were aligned with what customers really wanted, our experiments were much more likely to change their behavior for the better.

This pattern would repeat time and again, from the days when we were making less than a thousand dollars in revenue per month all the way up to the time we were making millions. In fact, this is the sign of a successful pivot: the new experiments you run are overall more productive than the experiments you were running before.

This is the pattern: poor quantitative results force us to declare failure and create the motivation, context, and space for more qualitative research. These investigations produce new ideas—new hypotheses—to be tested, leading to a possible pivot. Each pivot unlocks new opportunities for further experimentation, and the cycle repeats. Each time we repeat this simple rhythm: establish the baseline, tune the engine, and make a decision to pivot or persevere.

## **OPTIMIZATION VERSUS LEARNING**

Engineers, designers, and marketers are all skilled at optimization. For example, direct marketers are experienced at split testing value propositions by sending a different offer to two similar groups of customers so that they can measure differences in the response rates of the two groups. Engineers, of course, are skilled at improving a product's performance, just as designers are talented at making products easier to use. All these activities in a well-run traditional organization offer incremental benefit for incremental effort. As long as we are executing the plan well, hard work yields results.

However, these tools for product improvement do not work the same way for startups. If you are building the wrong thing, optimizing the product or its marketing will not yield significant results. A startup has to measure progress against a high bar: evidence that a sustainable business can be built around its products or services. That's a standard that can be assessed only if a startup has made clear, tangible predictions ahead of time.

In the absence of those predictions, product and strategy decisions are far more difficult and time-consuming. I often see this in my consulting practice. I've been called in many times to help a startup that feels that its engineering team "isn't working hard enough." When I meet with those teams, there are always improvements to be made and I recommend them, but invariably the real problem is not a lack of development talent, energy, or effort. Cycle after cycle, the team is working hard, but the business is not seeing results. Managers trained in a traditional model draw the logical conclusion: our team is not working hard, not working effectively, or not working efficiently.

Thus the downward cycle begins: the product development team valiantly tries to build a product according to the specifications it is receiving from the creative or business leadership. When good results are not forthcoming, business leaders assume that any discrepancy between what was planned and what was built is the cause and try to specify the next iteration in greater detail. As the specifications get more detailed, the planning process slows down, batch size increases, and feedback is delayed. If a board of directors or CFO is involved as a stakeholder, it doesn't take long for personnel changes to follow.

A few years ago, a team that sells products to large media companies invited me to help them as a consultant because they were concerned that their engineers were not working hard enough. However, the fault was not in the engineers; it was in the process the whole company was using to make decisions. They had customers but did not know them very well. They were deluged with feature requests from customers, the internal sales team, and the business leadership. Every new insight became an emergency that had to be addressed immediately. As a result, long-term projects were hampered by constant interruptions. Even worse, the team had no clear sense of whether any of the changes they were making mattered to customers. Despite the constant tuning and tweaking, the business results were consistently mediocre.

Learning milestones prevent this negative spiral by emphasizing a more likely possibility: the company is executing—with discipline!

—a plan that does not make sense. The innovation accounting framework makes it clear when the company is stuck and needs to change direction.

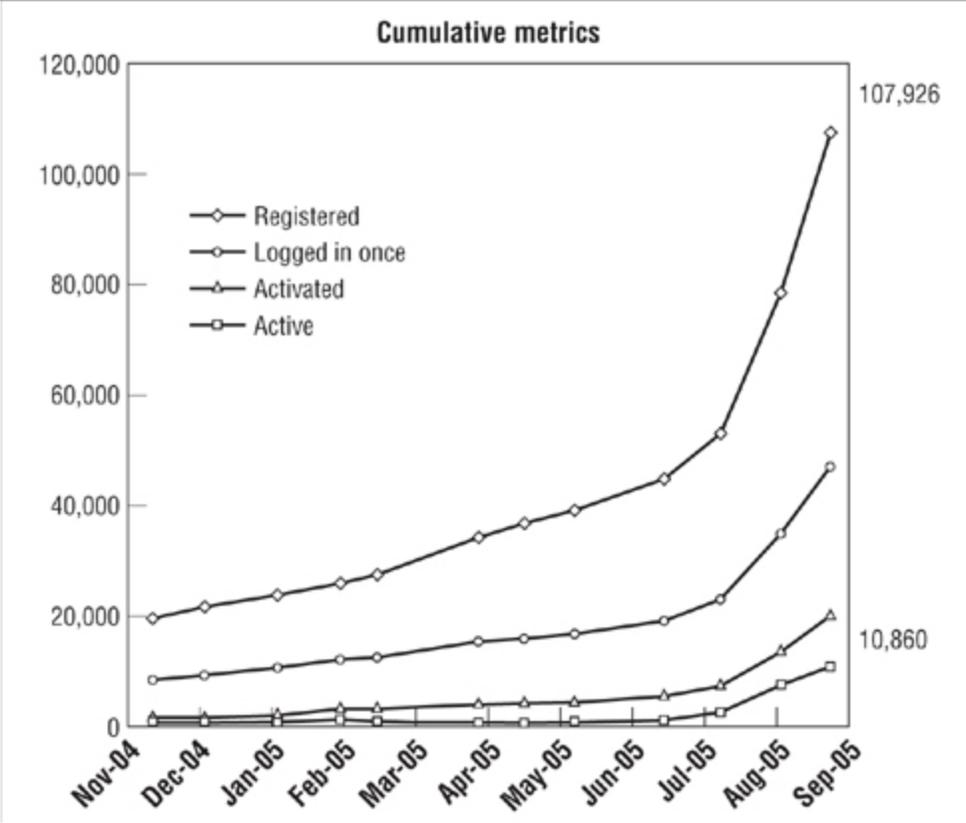
In the example above, early in the company's life, the product development team was incredibly productive because the company's founders had identified a large unmet need in the target market. The initial product, while flawed, was popular with early adopters. Adding the major features that customers asked for seemed to work wonders, as the early adopters spread the word about the innovation far and wide. But unasked and unanswered were other lurking questions: Did the company have a working engine of growth? Was this early success related to the daily work of the product development team? In most cases, the answer was no; success was driven by decisions the team had made in the past. None of its current initiatives were having any impact. But this was obscured because the company's gross metrics were all "up and to the right."

As we'll see in a moment, this is a common danger. Companies of any size that have a working engine of growth can come to rely on the wrong kind of metrics to guide their actions. This is what tempts managers to resort to the usual bag of success theater tricks: last-minute ad buys, channel stuffing, and whiz-bang demos, in a desperate attempt to make the gross numbers look better. Energy invested in success theater is energy that could have been used to help build a sustainable business. I call the traditional numbers used to judge startups "vanity metrics," and innovation accounting requires us to avoid the temptation to use them.

## **VANITY METRICS: A WORD OF CAUTION**

To see the danger of vanity metrics clearly, let's return once more to the early days of IMVU. Take a look at the following graph, which is from the same era in IMVU's history as that shown earlier in this chapter. It covers the same time period as the cohort-style graph on [this page](#); in fact, it is from the same board presentation.

This graph shows the traditional gross metrics for IMVU so far: total registered users and total paying customers (the gross revenue graph looks almost the same). From this viewpoint, things look much more exciting. That's why I call these vanity metrics: they give the rosiest possible picture. You'll see a traditional hockey stick graph (the ideal in a rapid-growth company). As long as you focus on the top-line numbers (signing up more customers, an increase in overall revenue), you'll be forgiven for thinking this product development team is making great progress. The company's growth engine is working. Each month it is able to acquire customers and has a positive return on investment. The excess revenue from those customers is reinvested the next month in acquiring more. That's where the growth is coming from.



But think back to the same data presented in a cohort style. IMVU is adding new customers, but it is not improving the yield on each new group. The engine is turning, but the efforts to tune the engine are not bearing much fruit. From the traditional graph alone, you

cannot tell whether IMVU is on pace to build a sustainable business; you certainly can't tell anything about the efficacy of the entrepreneurial team behind it.

Innovation accounting will not work if a startup is being misled by these kinds of vanity metrics: gross number of customers and so on. The alternative is the kind of metrics we use to judge our business and our learning milestones, what I call *actionable metrics*.

## **ACTIONABLE METRICS VERSUS VANITY METRICS**

To get a better sense of the importance of good metrics, let's look at a company called Grockit. Its founder, Farbood Nivi, spent a decade working as a teacher at two large for-profit education companies, Princeton Review and Kaplan, helping students prepare for standardized tests such as the GMAT, LSAT, and SAT. His engaging classroom style won accolades from his students and promotions from his superiors; he was honored with Princeton Review's National Teacher of the Year award. But Farb was frustrated with the traditional teaching methods used by those companies. Teaching six to nine hours per day to thousands of students, he had many opportunities to experiment with new approaches.<sup>2</sup>

Over time, Farb concluded that the traditional lecture model of education, with its one-to-many instructional approach, was inadequate for his students. He set out to develop a superior approach, using a combination of teacher-led lectures, individual homework, and group study. In particular, Farb was fascinated by how effective the student-to-student peer-driven learning method was for his students. When students could help each other, they benefited in two ways. First, they could get customized instruction from a peer who was much less intimidating than a teacher. Second, they could reinforce their learning by teaching it to others. Over time, Farb's classes became increasingly social—and successful.

As this unfolded, Farb felt more and more that his physical presence in the classroom was less important. He made an important connection: "I have this social learning model in my classroom. There's all this social stuff going on on the web." His idea was to

bring social peer-to-peer learning to people who could not afford an expensive class from Kaplan or Princeton Review or an even more expensive private tutor. From this insight Grockit was born.

Farb explains, “Whether you’re studying for the SAT or you’re studying for algebra, you study in one of three ways. You spend some time with experts, you spend some time on your own, and you spend some time with your peers. Grockit offers these three same formats of studying. What we do is we apply technology and algorithms to optimize those three forms.”

Farb is the classic entrepreneurial visionary. He recounts his original insight this way: “Let’s forget educational design up until now, let’s forget what’s possible and just redesign learning with today’s students and today’s technology in mind. There were plenty of multi-billion-dollar organizations in the education space, and I don’t think they were innovating in the way that we needed them to and I didn’t think we needed them anymore. To me, it’s really all about the students and I didn’t feel like the students were being served as well as they could.”

Today Grockit offers many different educational products, but in the beginning Farb followed a lean approach. Grockit built a minimum viable product, which was simply Farb teaching test prep via the popular online web conferencing tool WebEx. He built no custom software, no new technology. He simply attempted to bring his new teaching approach to students via the Internet. News about a new kind of private tutoring spread quickly, and within a few months Farb was making a decent living teaching online, with monthly revenues of \$10,000 to \$15,000. But like many entrepreneurs with ambition, Farb didn’t build his MVP just to make a living. He had a vision of a more collaborative, more effective kind of teaching for students everywhere. With his initial traction, he was able to raise money from some of the most prestigious investors in Silicon Valley.

When I first met Farb, his company was already on the fast track to success. They had raised venture capital from well-regarded investors, had built an awesome team, and were fresh off an

impressive debut at one of Silicon Valley's famous startup competitions.

They were extremely process-oriented and disciplined. Their product development followed a rigorous version of the agile development methodology known as Extreme Programming (described below), thanks to their partnership with a San Francisco-based company called Pivotal Labs. Their early product was hailed by the press as a breakthrough.

There was only one problem: they were not seeing sufficient growth in the use of the product by customers. Grockit is an excellent case study because its problems were not a matter of failure of execution or discipline.

Following standard agile practice, Grockit's work proceeded in a series of *sprints*, or one-month iteration cycles. For each sprint, Farb would prioritize the work to be done that month by writing a series of *user stories*, a technique taken from agile development. Instead of writing a specification for a new feature that described it in technical terms, Farb would write a story that described the feature from the point of view of the customer. That story helped keep the engineers focused on the customer's perspective throughout the development process.

Each feature was expressed in plain language in terms everyone could understand whether they had a technical background or not. Again following standard agile practice, Farb was free to reprioritize these stories at any time. As he learned more about what customers wanted, he could move things around in the *product backlog*, the queue of stories yet to be built. The only limit on this ability to change directions was that he could not interrupt any task that was in progress. Fortunately, the stories were written in such a way that the batch size of work (which I'll discuss in more detail in [Chapter 9](#)) was only a day or two.

This system is called agile development for a good reason: teams that employ it are able to change direction quickly, stay light on their feet, and be highly responsive to changes in the business requirements of the product owner (the manager of the process—in this case Farb—who is responsible for prioritizing the stories).

How did the team feel at the end of each sprint? They consistently delivered new product features. They would collect feedback from customers in the form of anecdotes and interviews that indicated that at least some customers liked the new features. There was always a certain amount of data that showed improvement: perhaps the total number of customers was increasing, the total number of questions answered by students was going up, or the number of returning customers was increasing.

However, I sensed that Farb and his team were left with lingering doubts about the company's overall progress. Was the increase in their numbers actually caused by their development efforts? Or could it be due to other factors, such as mentions of Grockit in the press? When I met the team, I asked them this simple question: How do you know that the prioritization decisions that Farb is making actually make sense?

Their answer: "That's not our department. Farb makes the decisions; we execute them."

At that time Grockit was focused on just one customer segment: prospective business school students who were studying for the GMAT. The product allowed students to engage in online study sessions with fellow students who were studying for the same exam. The product was working: the students who completed their studying via Grockit achieved significantly higher scores than they had before. But the Grockit team was struggling with the age-old startup problems: How do we know which features to prioritize? How can we get more customers to sign up and pay? How can we get out the word about our product?

I put this question to Farb: "How confident are you that you are making the right decisions in terms of establishing priorities?" Like most startup founders, he was looking at the available data and making the best educated guesses he could. But this left a lot of room for ambiguity and doubt.

Farb believed in his vision thoroughly and completely, yet he was starting to question whether his company was on pace to realize that vision. The product improved every day, but Farb wanted to make sure those improvements mattered to customers. I believe he

deserves a lot of credit for realizing this. Unlike many visionaries, who cling to their original vision no matter what, Farb was willing to put his vision to the test.

Farb worked hard to sustain his team's belief that Grockit was destined for success. He was worried that morale would suffer if anyone thought that the person steering the ship was uncertain about which direction to go. Farb himself wasn't sure if his team would embrace a true learning culture. After all, this was part of the grand bargain of agile development: engineers agree to adapt the product to the business's constantly changing requirements but are not responsible for the quality of those business decisions.

Agile is an efficient system of development from the point of view of the developers. It allows them to stay focused on creating features and technical designs. An attempt to introduce the need to learn into that process could undermine productivity.

(Lean manufacturing faced similar problems when it was introduced in factories. Managers were used to focusing on the utilization rate of each machine. Factories were designed to keep machines running at full capacity as much of the time as possible. Viewed from the perspective of the machine, that is efficient, but from the point of view of the productivity of the entire factory, it is wildly inefficient at times. As they say in systems theory, that which optimizes one part of the system necessarily undermines the system as a whole.)

What Farb and his team didn't realize was that Grockit's progress was being measured by vanity metrics: the total number of customers and the total number of questions answered. That was what was causing his team to spin its wheels; those metrics gave the team the sensation of forward motion even though the company was making little progress. What's interesting is how closely Farb's method followed superficial aspects of the Lean Startup learning milestones: they shipped an early product and established some baseline metrics. They had relatively short iterations, each of which was judged by its ability to improve customer metrics.

However, because Grockit was using the wrong kinds of metrics, the startup was not genuinely improving. Farb was frustrated in his

efforts to learn from customer feedback. In every cycle, the type of metrics his team was focused on would change: one month they would look at gross usage numbers, another month registration numbers, and so on. Those metrics would go up and down seemingly on their own. He couldn't draw clear cause-and-effect inferences. Prioritizing work correctly in such an environment is extremely challenging.

Farb could have asked his data analyst to investigate a particular question. For example, when we shipped feature X, did it affect customer behavior? But that would have required tremendous time and effort. When, exactly, did feature X ship? Which customers were exposed to it? Was anything else launched around that same time? Were there seasonal factors that might be skewing the data? Finding these answers would have required parsing reams and reams of data. The answer often would come weeks after the question had been asked. In the meantime, the team would have moved on to new priorities and new questions that needed urgent attention.

Compared to a lot of startups, the Grockit team had a huge advantage: they were tremendously disciplined. A disciplined team may apply the wrong methodology but can shift gears quickly once it discovers its error. Most important, a disciplined team can experiment with its own working style and draw meaningful conclusions.

## **Cohorts and Split-tests**

Grockit changed the metrics they used to evaluate success in two ways. Instead of looking at gross metrics, Grockit switched to cohort-based metrics, and instead of looking for cause-and-effect relationships after the fact, Grockit would launch each new feature as a true split-test experiment.

A split-test experiment is one in which different versions of a product are offered to 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. This technique was pioneered by direct mail advertisers. For example,

consider a company that sends customers a catalog of products to buy, such as Lands' End or Crate & Barrel. If you wanted to test a catalog design, you could send a new version of it to 50 percent of the customers and send the old standard catalog to the other 50 percent. To assure a scientific result, both catalogs would contain identical products; the only difference would be the changes to the design. To figure out if the new design was effective, all you would have to do was keep track of the sales figures for both groups of customers. (This technique is sometimes called A/B testing after the practice of assigning letter names to each variation.) Although split testing often is thought of as a marketing-specific (or even a direct marketing-specific) practice, Lean Startups incorporate it directly into product development.

These changes led to an immediate change in Farb's understanding of the business. Split testing often uncovers surprising things. For example, many features that make the product better in the eyes of engineers and designers have no impact on customer behavior. This was the case at Grockit, as it has been in every company I have seen adopt this technique. Although working with split tests seems to be more difficult because it requires extra accounting and metrics to keep track of each variation, it almost always saves tremendous amounts of time in the long run by eliminating work that doesn't matter to customers.

Split testing also helps teams refine their understanding of what customers want and don't want. Grockit's team constantly added new ways for their customers to interact with each other in the hope that those social communication tools would increase the product's value. Inherent in those efforts was the belief that customers desired more communication during their studying. When split testing revealed that the extra features did not change customer behavior, it called that belief into question.

The questioning inspired the team to seek a deeper understanding of what customers really wanted. They brainstormed new ideas for product experiments that might have more impact. In fact, many of these ideas were not new. They had simply been overlooked because the company was focused on building social tools. As a result,

Grockit tested an intensive solo-studying mode, complete with quests and gamelike levels, so that students could have the choice of studying by themselves or with others. Just as in Farb's original classroom, this proved extremely effective. Without the discipline of split testing, the company might not have had this realization. In fact, over time, through dozens of tests, it became clear that the key to student engagement was to offer them a combination of social and solo features. Students preferred having a choice of how to study.

## **Kanban**

Following the lean manufacturing principle of *kanban*, or capacity constraint, Grockit changed the product prioritization process. Under the new system, user stories were not considered complete until they led to validated learning. Thus, stories could be cataloged as being in one of four states of development: in the product backlog, actively being built, done (feature complete from a technical point of view), or in the process of being validated. Validated was defined as "knowing whether the story was a good idea to have been done in the first place." This validation usually would come in the form of a split test showing a change in customer behavior but also might include customer interviews or surveys.

The *kanban* rule permitted only so many stories in each of the four states. As stories flow from one state to the other, the buckets fill up. Once a bucket becomes full, it cannot accept more stories. Only when a story has been validated can it be removed from the *kanban* board. If the validation fails and it turns out the story is a bad idea, the relevant feature is removed from the product (see the chart on [this page](#)).

### **KANBAN DIAGRAM OF WORK AS IT PROGRESSES FROM STAGE TO STAGE**

(No bucket can contain more than three projects at a time.)

BACKLOG	IN PROGRESS	BUILT	VALIDATED
A B C	D E	F	

Work on A begins. D and E are in development. F awaits validation.

BACKLOG	IN PROGRESS	BUILT	VALIDATED
G H I	B C	D E A	F

F is validated. D and E await validation. G, H, I are new tasks to be undertaken. B and C are being built. A completes development.

BACKLOG	IN PROGRESS	BUILT	VALIDATED
H → I →	G B → C →	D E A	F

B and C have been built, but under *kanban*, cannot be moved to the next bucket for validation until A, D, E have been validated. Work cannot begin on H and I until space opens up in the buckets ahead.

I have implemented this system with several teams, and the initial result is always frustrating: each bucket fills up, starting with the “validated” bucket and moving on to the “done” bucket, until it’s not possible to start any more work. Teams that are used to measuring their productivity narrowly, by the number of stories they are delivering, feel stuck. The only way to start work on new features is to investigate some of the stories that are done but haven’t been validated. That often requires nonengineering efforts: talking to customers, looking at split-test data, and the like.

Pretty soon everyone gets the hang of it. This progress occurs in fits and starts at first. Engineering may finish a big batch of work, followed by extensive testing and validation. As engineers look for ways to increase their productivity, they start to realize that if they include the validation exercise from the beginning, the whole team can be more productive.

For example, why build a new feature that is not part of a split-test experiment? It may save you time in the short run, but it will take more time later to test, during the validation phase. The same logic applies to a story that an engineer doesn't understand. Under the old system, he or she would just build it and find out later what it was for. In the new system, that behavior is clearly counterproductive: without a clear hypothesis, how can a story ever be validated? We saw this behavior at IMVU, too. I once saw a junior engineer face down a senior executive over a relatively minor change. The engineer insisted that the new feature be split-tested, just like any other. His peers backed him up; it was considered absolutely obvious that all features should be routinely tested, no matter who was commissioning them. (Embarrassingly, all too often I was the executive in question.) A solid process lays the foundation for a healthy culture, one where ideas are evaluated by merit and not by job title.

Most important, teams working in this system begin to measure their productivity according to validated learning, not in terms of the production of new features.

## **Hypothesis Testing at Grockit**

When Grockit made this transition, the results were dramatic. In one case, they decided to test one of their major features, called lazy registration, to see if it was worth the heavy investment they were making in ongoing support. They were confident in this feature because lazy registration is considered one of the design best practices for online services. In this system, customers do not have to register for the service up front. Instead, they immediately begin

using the service and are asked to register only after they have had a chance to experience the service's benefit.

For a student, lazy registration works like this: when you come to the Grockit website, you're immediately placed in a study session with other students working on the same test. You don't have to give your name, e-mail address, or credit card number. There is nothing to prevent you from jumping in and getting started immediately. For Grockit, this was essential to testing one of its core assumptions: that customers would be willing to adopt this new way of learning only if they could see proof that it was working early on.

As a result of this hypothesis, Grockit's design required that it manage three classes of users: unregistered guests, registered (trial) guests, and customers who had paid for the premium version of the product. This design required significant extra work to build and maintain: the more classes of users there are, the more work is required to keep track of them, and the more marketing effort is required to create the right incentives to entice customers to upgrade to the next class. Grockit had undertaken this extra effort because lazy registration was considered an industry best practice.

I encouraged the team to try a simple split-test. They took one cohort of customers and required that they register immediately, based on nothing more than Grockit's marketing materials. To their surprise, this cohort's behavior was exactly the same as that of the lazy registration group: they had the same rate of registration, activation, and subsequent retention. In other words, the extra effort of lazy registration was a complete waste even though it was considered an industry best practice.

Even more important than reducing waste was the insight that this test suggested: customers were basing their decision about Grockit on something other than their use of the product.

Think about this. Think about the cohort of customers who were required to register for the product before entering a study session with other students. They had very little information about the product, nothing more than was presented on Grockit's home page and registration page. By contrast, the lazy registration group had a tremendous amount of information about the product because they

had used it. Yet despite this information disparity, customer behavior was exactly the same.

This suggested that improving Grockit's positioning and marketing might have a more significant impact on attracting new customers than would adding new features. This was just the first of many important experiments Grockit was able to run. Since those early days, they have expanded their customer base dramatically: they now offer test prep for numerous standardized tests, including the GMAT, SAT, ACT, and GRE, as well as online math and English courses for students in grades 7 through 12.

Grockit continues to evolve its process, seeking continuous improvement at every turn. With more than twenty employees in its San Francisco office, Grockit continues to operate with the same deliberate, disciplined approach that has been their hallmark all along. They have helped close to a million students and are sure to help millions more.

## **THE VALUE OF THE THREE A'S**

These examples from Grockit demonstrate each of the three A's of metrics: actionable, accessible, and auditable.

### **Actionable**

For a report to be considered actionable, it must demonstrate clear cause and effect. Otherwise, it is a vanity metric. The reports that Grockit's team began to use to judge their learning milestones made it extremely clear what actions would be necessary to replicate the results.

By contrast, vanity metrics fail this criterion. Take the number of hits to a company website. Let's say we have 40,000 hits this month—a new record. What do we need to do to get more hits? Well, that depends. Where are the new hits coming from? Is it from 40,000 new customers or from one guy with an extremely active web browser? Are the hits the result of a new marketing campaign or PR push? What is a hit, anyway? Does each page in the browser count

as one hit, or do all the embedded images and multimedia content count as well? Those who have sat in a meeting debating the units of measurement in a report will recognize this problem.

Vanity metrics wreak havoc because they prey on a weakness of the human mind. In my experience, when the numbers go up, people think the improvement was caused by their actions, by whatever they were working on at the time. That is why it's so common to have a meeting in which marketing thinks the numbers went up because of a new PR or marketing effort and engineering thinks the better numbers are the result of the new features it added. Finding out what is actually going on is extremely costly, and so most managers simply move on, doing the best they can to form their own judgment on the basis of their experience and the collective intelligence in the room.

Unfortunately, when the numbers go down, it results in a very different reaction: now it's somebody else's fault. Thus, most team members or departments live in a world where their department is constantly making things better, only to have their hard work sabotaged by other departments that just don't get it. Is it any wonder these departments develop their own distinct language, jargon, culture, and defense mechanisms against the bozos working down the hall?

Actionable metrics are the antidote to this problem. When cause and effect is clearly understood, people are better able to learn from their actions. Human beings are innately talented learners when given a clear and objective assessment.

## **Accessible**

All too many reports are not understood by the employees and managers who are supposed to use them to guide their decision making. Unfortunately, most managers do not respond to this complexity by working hand in hand with the data warehousing team to simplify the reports so that they can understand them better. Departments too often spend their energy learning how to

use data to get what they want rather than as genuine feedback to guide their future actions.

There is an antidote to this misuse of data. First, make the reports as simple as possible so that everyone understands them. Remember the saying “Metrics are people, too.” The easiest way to make reports comprehensible is to use tangible, concrete units. What is a website hit? Nobody is really sure, but everyone knows what a person visiting the website is: one can practically picture those people sitting at their computers.

This is why cohort-based reports are the gold standard of learning metrics: they turn complex actions into people-based reports. Each cohort analysis says: among the people who used our product in this period, here’s how many of them exhibited each of the behaviors we care about. In the IMVU example, we saw four behaviors: downloading the product, logging into the product from one’s computer, engaging in a chat with other customers, and upgrading to the paid version of the product. In other words, the report deals with people and their actions, which are far more useful than piles of data points. For example, think about how hard it would have been to tell if IMVU was being successful if we had reported only on the total number of person-to-person conversations. Let’s say we have 10,000 conversations in a period. Is that good? Is that one person being very, very social, or is it 10,000 people each trying the product one time and then giving up? There’s no way to know without creating a more detailed report.

As the gross numbers get larger, accessibility becomes more and more important. It is hard to visualize what it means if the number of website hits goes down from 250,000 in one month to 200,000 the next month, but most people understand immediately what it means to lose 50,000 customers. That’s practically a whole stadium full of people who are abandoning the product.

Accessibility also refers to widespread access to the reports. Grockit did this especially well. Every day their system automatically generated a document containing the latest data for every single one of their split-test experiments and other leap-of-faith metrics. This document was mailed to every employee of the

company: they all always had a fresh copy in their e-mail in-boxes. The reports were well laid out and easy to read, with each experiment and its results explained in plain English.

Another way to make reports accessible is to use a technique we developed at IMVU. Instead of housing the analytics or data in a separate system, our reporting data and its infrastructure were considered part of the product itself and were owned by the product development team. The reports were available on our website, accessible to anyone with an employee account.

Each employee could log in to the system at any time, choose from a list of all current and past experiments, and see a simple one-page summary of the results. Over time, those one-page summaries became the de facto standard for settling product arguments throughout the organization. When people needed evidence to support something they had learned, they would bring a printout with them to the relevant meeting, confident that everyone they showed it to would understand its meaning.

## **Auditable**

When informed that their pet project is a failure, most of us are tempted to blame the messenger, the data, the manager, the gods, or anything else we can think of. That's why the third A of good metrics, "auditable," is so essential. We must ensure that the data is credible to employees.

The employees at IMVU would brandish one-page reports to demonstrate what they had learned to settle arguments, but the process often wasn't so smooth. Most of the time, when a manager, developer, or team was confronted with results that would kill a pet project, the loser of the argument would challenge the veracity of the data.

Such challenges are more common than most managers would admit, and unfortunately, most data reporting systems are not designed to answer them successfully. Sometimes this is the result of a well-intentioned but misplaced desire to protect the privacy of customers. More often, the lack of such supporting documentation is

simply a matter of neglect. Most data reporting systems are not built by product development teams, whose job is to prioritize and build product features. They are built by business managers and analysts. Managers who must use these systems can only check to see if the reports are mutually consistent. They all too often lack a way to test if the data is consistent with reality.

The solution? First, remember that “Metrics are people, too.” We need to be able to test the data by hand, in the messy real world, by talking to customers. This is the only way to be able to check if the reports contain true facts. Managers need the ability to spot check the data with real customers. It also has a second benefit: systems that provide this level of auditability give managers and entrepreneurs the opportunity to gain insights into why customers are behaving the way the data indicate.

Second, those building reports must make sure the mechanisms that generate the reports are not too complex. Whenever possible, reports should be drawn directly from the master data, rather than from an intermediate system, which reduces opportunities for error. I have noticed that every time a team has one of its judgments or assumptions overturned as a result of a technical problem with the data, its confidence, morale, and discipline are undermined.



When we watch entrepreneurs succeed in the mythmaking world of Hollywood, books, and magazines, the story is always structured the same way. First, we see the plucky protagonist having an epiphany, hatching a great new idea. We learn about his or her character and personality, how he or she came to be in the right place at the right time, and how he or she took the dramatic leap to start a business.

Then the photo montage begins. It’s usually short, just a few minutes of time-lapse photography or narrative. We see the protagonist building a team, maybe working in a lab, writing on whiteboards, closing sales, pounding on a few keyboards. At the end of the montage, the founders are successful, and the story can move on to more interesting fare: how to split the spoils of their success,

who will appear on magazine covers, who sues whom, and implications for the future.

Unfortunately, the real work that determines the success of startups happens during the photo montage. It doesn't make the cut in terms of the big story because it is too boring. Only 5 percent of entrepreneurship is the big idea, the business model, the whiteboard strategizing, and the splitting up of the spoils. The other 95 percent is the gritty work that is measured by innovation accounting: product prioritization decisions, deciding which customers to target or listen to, and having the courage to subject a grand vision to constant testing and feedback.

One decision stands out above all others as the most difficult, the most time-consuming, and the biggest source of waste for most startups. We all must face this fundamental test: deciding when to pivot and when to persevere. To understand what happens during the photo montage, we have to understand how to pivot, and that is the subject of [Chapter 8](#).