

Chapter 16. Stage Two: Stickiness

Having climbed inside your market's head, it's time to build something. The big question now is whether or not what you've built is sticky, so that when you throw users at it, they'll engage. You want to be, as Rowan Atkinson's *Blackadder* put it, "in the stickiest situation since Sticky the stick insect got stuck on a sticky bun." That's how you make the business sustainable.

MVP Stickiness

The focus now is squarely on retention and engagement. You can look at daily, weekly, and/or monthly active users; how long it takes someone to become inactive; how many inactive users can be reactivated when sent an email; and which features engaged users spend time with, and which they ignore. Segment these metrics by cohort to see if your changes convince additional users to behave differently. Did users who signed up in February stick around longer than those who joined in January?

You don't just want signs of engagement. You want *proof* that your product is becoming an integral part of your users' lives, and that it'll be hard for them to switch. You're not looking for, nor should you expect, rapid growth. You're throwing things at the wall to test stickiness, not measuring how fast you can throw. And by "things," we mean users. After all, if you can't convince a hundred users to stick around today, you're unlikely to convince a million to do so later.^[57]

Your top priority is to build a core set of features that gets used regularly and successfully, even by a small group of initial users. Without that, you don't have a solid enough foundation for growth. Your initial target market can be very small, hyper-focused on the smallest subset of users that you think will generate meaningful results.

Ultimately, you need to prove two things before you can move on to the Virality stage:

- Are people using the product as you expected? If they aren't, maybe you should switch to that new use case or market, as PayPal did when it changed from PalmPilot to web-based payment or when Autodesk stopped making desktop automation and instead focused on design tools.
- Are people getting enough value out of it? They may like it, but if they won't pay, click ads, or invite their friends, you may not have a business.

Don't drive new traffic until you know you can turn that extra attention into engagement. When you know users keep coming back, it's time to grow your user base.

Iterating the MVP

As we've said, the MVP is a process, not a product. You don't pass Go just because you put something into people's hands. Expect to go through many iterations of your MVP before it's time to shift your focus to customer acquisition.

Iterating on your MVP is difficult, tedious work. It's methodical. Sometimes it doesn't feel like innovation. Iterations are evolutionary; pivots are revolutionary. This is one of the reasons founders get frustrated and decide instead to pivot repeatedly in the hopes that something will accidentally engage their users. Resist that temptation.

As you iterate, your goal is to improve on the core metrics that you're tracking. If a new feature doesn't significantly improve the One Metric That Matters, remove it. Don't get caught tinkering and polishing. You're not fine-tuning at this point; you're searching for the right product and market.

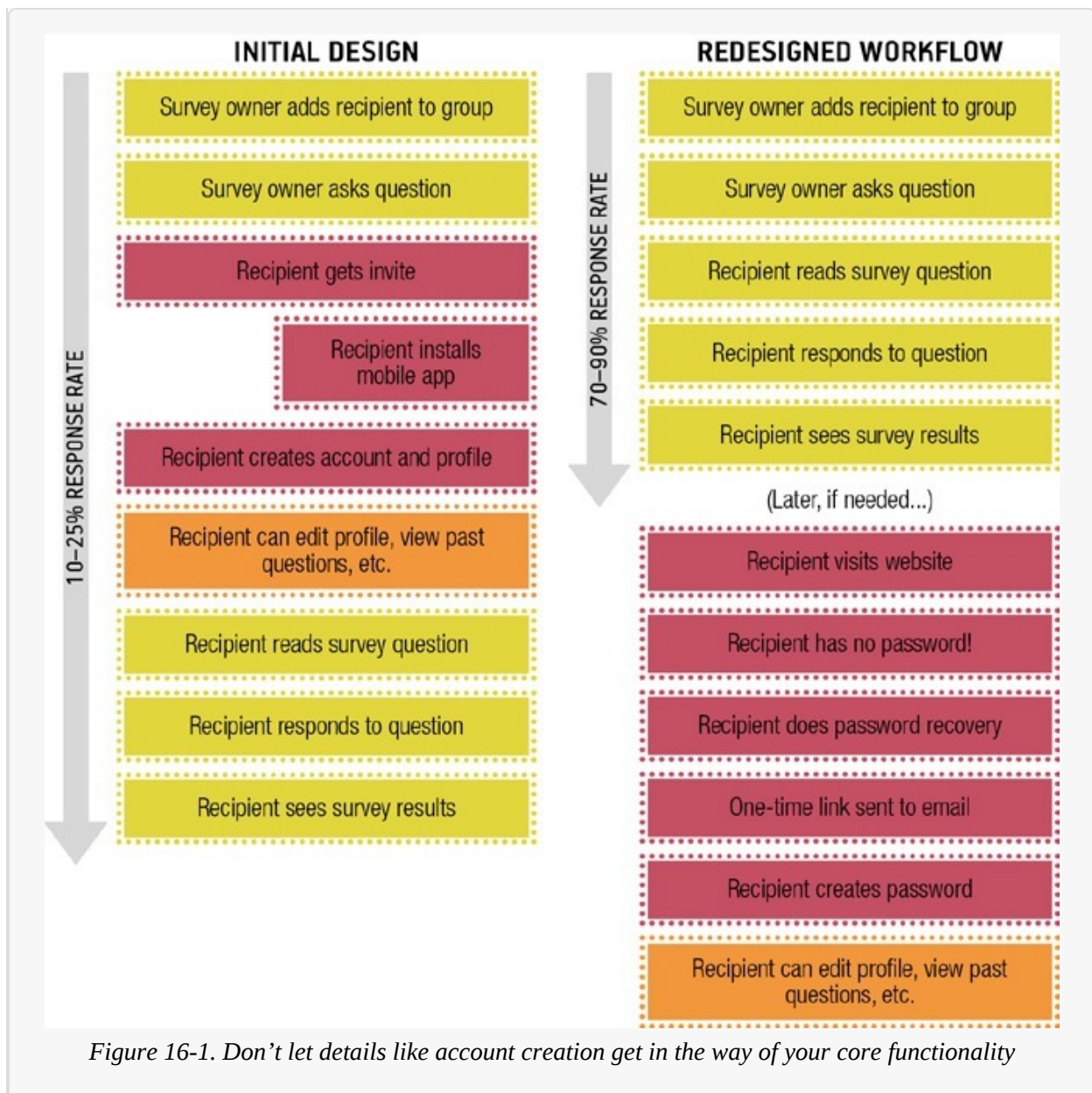
qidiq Changes How It Adds Users

Qidiq is a tool—for doing really simple surveys of small groups via email or a mobile application—that was launched through startup accelerator Year One Labs. In early versions of the product, a survey creator invited respondents to

join a group. Once those respondents had signed up and created an account, they could answer surveys delivered by email or pushed to an iPhone client.

Only a small percentage of people who were invited actually created an account and responded. So the founders devised a test: why not act as if the recipient already had an account, send her a survey question she can respond to with a single click or tap, and see what the response rate is like? The act of responding could be treated as tacit acceptance of enrollment; later, if the recipient wanted to log into her account, she could do so through a password recovery.

The qidiq team quickly changed their application, as illustrated in **Figure 16-1**, and sent out more surveys to personal groups they'd created. These initial surveys were sent via email alone. The results were striking: response rates went from 10–25% with the enroll-first model to 70–90% with the vote-first model. This made the team rethink their plans to develop a mobile application, since mobile applications couldn't compete with the cross-platform ubiquity and immediacy of email. Maybe email was good enough, and they shouldn't build their mobile app any more, or port it to Android.



“By focusing on the key metric of response rate, we were able to avoid the temptation of wasting our energy on the sexier mobile app,” says co-founder Jonathan Abrams. “Because it was the response rate that mattered, it became clear early on that email, while less sexy, was the better strategy for our startup.” The metric qidiq was tracking, which was the basis of its whole product, was the number of people who would respond to a question. That was the right metric, and when the team found a product change that moved it dramatically in the right direction, it made them rethink the design of their entire service.

Summary

- The MVP should include the simplest, least-friction path between your user and the “aha!” moment you’re trying to deliver.
- Everything is on the table. While you shouldn’t reinvent well-understood concepts like an enrollment process with which people are familiar, you should also feel free to ignore them for the sake of a test.
- Focusing on a single metric—in this case, survey response rate—let the team tweak every other part of the business, from sign-up to platform.

Analytics Lessons Learned

When you’ve got an MVP, you don’t have a product. You have a tool for figuring out what product to build. By asking an unorthodox question—in this case, “What if users were already registered?”—the qidiq team not only quadrupled response rates, but also avoided a costly, distracting development rathole.

Premature Virality

Many startups—particularly in the consumer space—focus on virality first. They implement features and tactics to try to increase user acquisition as much as possible, before really understanding what those users will do. This is common for two reasons:

- First, the bar for success in a consumer application is always going up. A few years ago, hundreds of thousands of users was considered big. Today, 1 million users is the benchmark, but it’s quickly going to 10 million. That’s a lot of users. Certain categories of product, such as social networks and e-commerce, are ossifying, with a few gigantic players competing and leaving little room for upstarts.
- Second, many consumer applications rely on network effects. The more users, the more value created for everyone. Nobody wants to use the telephone when they’re the only one with a telephone. Location-based applications typically require lots of scale, as do most marketplaces and user-

generated content businesses, so that there are enough transactions and discussions to make things interesting. Without a critical mass of users, Facebook is an empty shell. Reaching this critical mass quickly is the first step in delivering the anticipated value of the product.

As a result, founders of consumer startups and multiplayer games often argue that they need to focus on virality and user acquisition because it will solve all their other problems. But having lots of users isn't traction unless those users are engaged and sticking around.

The results of premature scaling can be disastrous if startups invest all of their time and money into user acquisition, only to watch those users churn too quickly. By the time they go back and try to recover those users, they're gone. You never get a second chance to have a first enrollment.

The Goal Is Retention

The more engaged that people are with your product (and potentially other users of your product), the more likely they'll stay. By ignoring growth from virality (for now), you can simplify how you decide what to build next into your MVP. Ask yourself, "Do we believe that the feature we want to build (or the feature we want to change) will improve stickiness?" Put the feature aside if the answer is "no." But if the answer is "yes," figure out how to test that belief and start building the feature.

Seven Questions to Ask Yourself Before Building a Feature

You probably have a long list of feature ideas you believe will improve retention. You need to further prioritize. Here are seven questions you can ask yourself (and your team) before building a new feature.

1. Why Will It Make Things Better?

You can't build a feature without having a reason for building it. In the Stickiness stage, your focus is retention. Look at your potential feature list and ask yourself, "Why do I think this will improve retention?"

You'll be tempted to copy what others are doing—say, using gamification to drive engagement (and in turn retention)—just because it looks like it's working for the competition. Don't. Qidiq ignored common wisdom around the sign-up process and the creation of a mobile app and quadrupled engagement. It's OK to copy existing patterns, but know why you're doing so.

Asking “Why will it make it better?” forces you to write out (on paper!) a hypothesis. This naturally leads to a good experiment that will test that hypothesis. Feature experiments, if they're tied to a specific metric (such as retention) are usually easy: you believe feature X will improve retention by Y percent. The second part of that statement is as important as the first part; you need to draw that line in the sand.

2. Can You Measure the Effect of the Feature?

Feature experiments require that you measure the impact of the feature. That impact has to be quantifiable. Too often, features get added to a product without any quantifiable validation—which is a direct path toward scope creep and feature bloat.

If you're unable to quantify the impact of a new feature, you can't assess its value, and you won't really know what to do with the feature over time. If this is the case, leave it as is, iterate on it, or kill it.

3. How Long Will the Feature Take to Build?

Time is a precious resource you never get back. You have to compare the relative development time of each feature on your list. If something is going to take months to build, you need good confidence that it will have a significant impact. Can you break it into smaller parts, or test the inherent risk with a curated MVP or a prototype instead?

4. Will the Feature Overcomplicate Things?

Complexity kills products. It's most obvious in the user experience of many web applications: they become so convoluted and confusing that users leave for a simpler alternative.

“And” is the enemy of success. When discussing a feature with your team, pay attention to how it's being described. “The feature will allow you to do this, *and* it'd be great if it did this other thing, *and* this other thing, *and* this other thing too.” Warning bells should be going off at this point. If you're trying to justify a

feature by saying it satisfies several needs a little bit, know that it's almost always better to satisfy *one* need in an absolutely epic, remarkable way.

One mobile analytics expert for an adult-content site told us his rule for new features is simple: "If you can't do it in three taps with one hand, it's broken." Knowing your user's behavior and expectations is everything. Having feature complexity get in the way of the real testing you need to do around your market, customer acquisition, and retention is extremely painful.

5. How Much Risk Is There in This New Feature?

Building new features always comes with some amount of risk. There's technical risk related to how a feature may impact the code base. There's user risk in terms of how people might respond to the feature. There's also risk in terms of how a new feature drives future development, potentially setting you on a path you don't want to pursue.

Each feature you add creates an emotional commitment for your development team, and sometimes for your customers. Analytics helps break that bond so that you can measure things honestly and make the best decisions possible, with the most information available.

6. How Innovative Is the New Feature?

Not everything you do will be innovative. Most features aren't innovative, they're small tweaks to a product in the hope that the whole is more valuable than the individual parts.

But consider innovation when prioritizing feature development; generally, the easiest things to do rarely have a big impact. You're still in the Stickiness stage, trying to find the right product. Changing a submit button from red to blue may result in a good jump in signup conversions (a classic A/B test), but it's probably not going to turn your business from a failure into a giant success; it's also easy for others to copy.

It's better to make big bets, swing for the fences, try more radical experiments, and build more disruptive things, particularly since you have fewer user expectations to contend with than you will later on.

7. What Do Users Say They Want?

Your users are important. Their feedback is important. But relying on what they say is risky. Be careful about over-prioritizing based on user input alone. Users

lie, and they don't like hurting your feelings.

Prioritizing feature development during an MVP isn't an exact science. User actions speak louder than words. Aim for a genuinely testable hypothesis for every feature you build, and you'll have a much better chance of quickly validating success or failure. Simply tracking how popular various features are within the application will reveal what's working and what's not. Looking at what feature a user was using before he hit "undo" or the back button will pinpoint possible problem areas.

Building features is easy if you plan them beforehand and truly understand why you're doing something. It's important to tie your high-level vision and long-term goals down to the feature level. Without that alignment, you run the risk of building features that can't be properly tested and don't drive the business forward.

How Rally Builds New Features with a Lean Approach

Rally Software makes Agile application lifecycle management software. The company was founded in 2002 and has pioneered a number of Agile best practices. We spoke with Chief Technologist Zach Nies about how the company continues to successfully build its products.

Establishing a Company Vision

Everything at Rally starts with a three-to five-year company vision that is refreshed every 18 months. The entire company aligns around the vision, which is the first waypoint in turning a big, distant goal into something more attainable. This longer-term vision becomes a key input into annual planning each year. Zach says, "When we were younger and smaller we didn't bother looking three years into the future, but it's an important part of the process for a company of our size."

Annual planning is initially done by a small group of executives. Zach calls this the first iteration. The output of the initial planning is a draft corporate strategy, which provides a clear, concise picture of Rally's performance gaps and targets, reflections, and rationale for the year. The executive team also identifies three or four high-level places where they believe the company needs to focus action to

four high-level places where they believe the company needs to focus action to accomplish the annual vision. “This work creates a draft of ideas to bring back to Rally for reflection,” Zach says. “They provide a summary of what the executive group saw as critically valuable to address in our upcoming year.”

The second iteration of annual planning takes the form of departmental annual retrospectives. Rally uses an approach called ORID (Objective, Reflective, Interpretive, Decisional) from *The Art of Focused Conversation* by R. Brian Stanfield (New Society Publishers).^[58] Zach says:

This process invites insights from all employees, and provides a valuable narrative about the past, present, and future. From each ORID within each department, we learn about completed work, the current work in progress, planned work, specific annual metrics, the implications for the coming year, and the overall mood for the year. Kids are learning machines, but adults need structured reflection to learn; this process provides that structure.

Both the executive planning and the ORIDs feed into the next step of the annual planning process: gathering 60 people from the company in a highly-facilitated meeting to clearly articulate the vision for the year and align around how to accomplish it.

Developing a Product Plan

The product team is actively involved in defining the company’s annual strategy. A big part of this is aligning the directions of the company and product. The product team focuses on answering the question “Why?” above everything else. “The articulation of why we’re doing something, and always questioning our focus, rallies everyone around one compelling vision, company, and product, and creates a vital emotional connection with our customers,” Zach says. “Only once we understand ‘why’ can we really look at ‘what’ and ‘how’.”

Now Rally is ready to dig into product. While this process may seem like a lot, it’s very iterative and Lean. The company goes through a *build → measure → learn* cycle at several levels before getting to the point where it’s actually developing features.

Deciding What to Build

Feature development begins in earnest with deciding what to build and how to build it. Rally has an open, but process-oriented, way of making feature decisions. Each quarter, employees submit short proposals for changes to the company’s product direction. These proposals come from anyone in the organization, but are typically highly influenced by interactions with customers.

Zach says:

We include almost everyone who does product-management-type work in the decision-making process, including product marketing, product owners, engineering managers, sales leadership, and executives. It may seem like this is quite a bit of process, but the benefits of everyone's input and alignment far outweigh the 10 or so hours a quarter we spend running the process. We find strong alignment enables great execution.

Rally doesn't release software, but instead "turns features on for users and customers." Most features have a toggle that allows Rally to turn them on or off for specific customers. This allows the company to roll out code to progressively larger groups of users, generating feedback from early adopters while mitigating the risk of exposing problems to a lot of customers.

Measuring Progress

Underneath Rally's feature development process, the company is focused on measurement. "We have an internal data warehouse in which we record everything from server/database kernel-level performance measurements to high-level user gestures derived from HTTP interactions between the browser and our servers," says Zach. The goal is to make sure the team can measure feature usage and performance. "When we develop a feature our product team can form theories about how much usage warrants further development of that feature," Zach says. "As we are toggling on the feature we can compare our theories to actual data. Because the data includes both usage and performance information, we can quickly understand, in real time, the impact a feature is having on the performance and stability of our production environment."

Learning Through Experiments

Even with such a deep level of planning and an all-inclusive approach to product development, Zach still says that the company is careful not to "blindly build features based on internal or customer requests." Instead, it runs experiments to learn more.

According to Zach, every experiment starts with a series of questions:

- What do we want to learn and why?
- What's the underlying problem we are trying to solve, and who is feeling the pain? This helps everyone involved have empathy for what we are doing.
- What's our hypothesis? This is written in the form: "[Specific repeatable

action] will create [expected result].” We make sure the hypothesis is written in such a way that the experiment is capable of invalidating it.

- How will we run the experiment, and what will we build to support it?
- Is the experiment safe to run?
- How will we conclude the experiment, and what steps will be taken to mitigate issues that result from the experiment’s conclusion?
- What measures will we use to invalidate our hypothesis with data? We also include what measures will indicate the experiment isn’t safe to continue.

In a three-month period, over 20 experiments were run to learn exactly what would satisfy users in a critical part of the user interface. Rather than guessing, this was a disciplined process of discovery. This area of the user interface was a focus because refining it was a major part of the product vision for the year, and directly supported one of Rally’s corporate goals for the year.

Summary

- Data-driven product direction starts at the top, and it’s an iterative, methodical process.
- Everything is an experiment, even when you have an established product and a loyal set of customers.
- It takes extra engineering effort to be able to turn on and off individual features, and to measure the resulting change in user behavior, but that investment pays off in reduced cycle time and better learning.

Analytics Lessons Learned

Rally has taken measurement to the next level. In a way, Rally is two companies—one making lifecycle management software, and one running a gigantic, continuous experiment on its users to better understand how they interact with the product itself. This requires a lot of discipline and focus, as well as considerable engineering effort to make every feature testable and measurable, but it’s paid off in less waste, a better product, and a consistent alignment with what customers want

How to Handle User Feedback

Customers have something in common with entrepreneurs—they're liars too. They don't lie intentionally, but often they forget how your product really works or what they were doing in the product.

Many of the reviews for personal banking app Mint give the product one star, saying, "Warning! This product will try to collect your banking information and connect to your bank account!" as shown in [Figure 16-2](#). *But that's what Mint is for.*

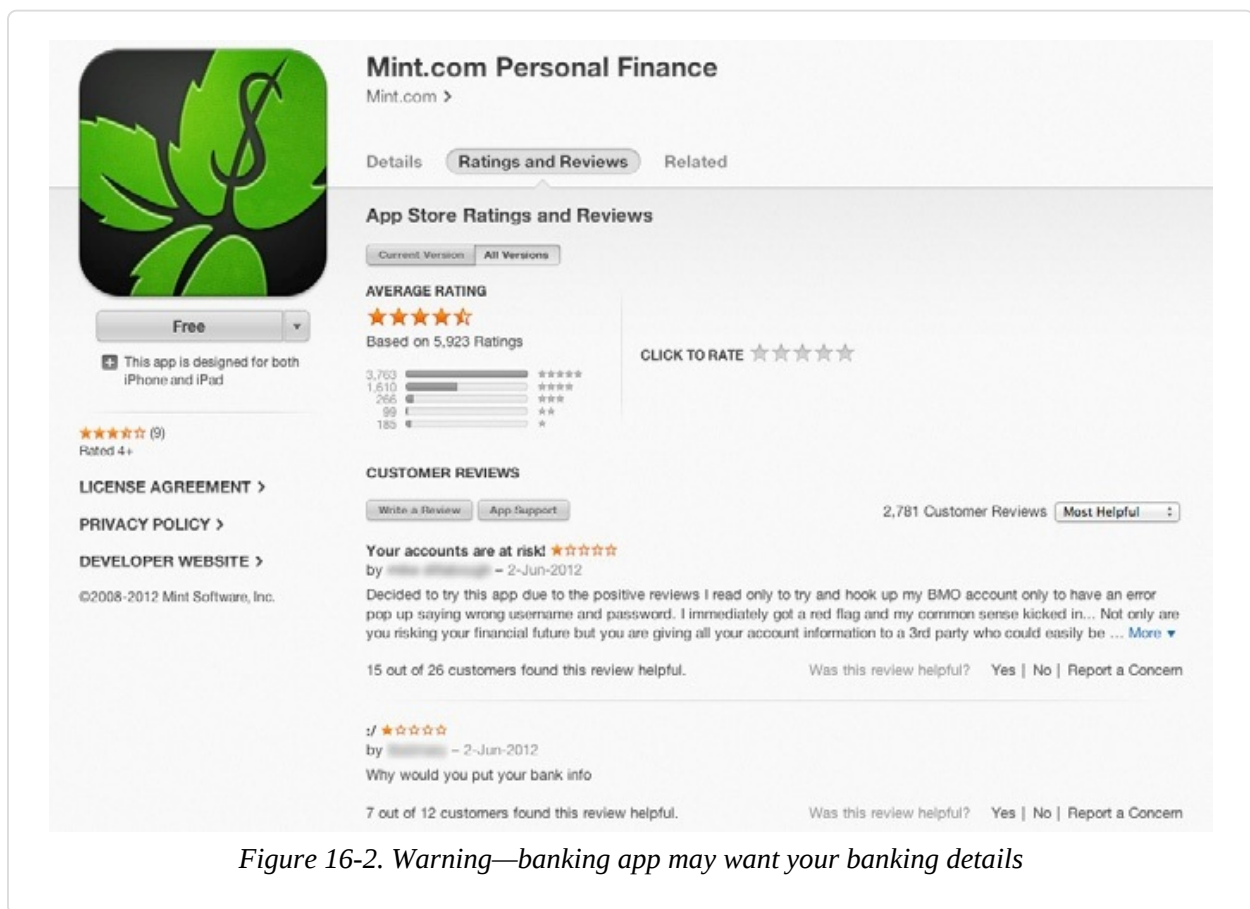


Figure 16-2. Warning—banking app may want your banking details

If you're the product manager, you might be tempted to ignore this feedback, but what it's really telling you is that your marketing and product descriptions aren't working, bringing down your product ratings and reducing your addressable

market.

Customers may give you feedback you don't like. Just remember that they don't have the same mental model you do, they aren't in your target market. They often lack training to use your product properly.

We've already seen some of the cognitive biases demonstrated by interview subjects. Existing users suffer from similar biases. They have different expectations and context from you. You need to view their feedback with that in mind.

For one thing, user feedback suffers from horrible sampling bias. Few people provide feedback when they have a predictable, tepid experience. They reach out when they're ecstatic or furious. If they're feeling aggrieved, you'll hear from them.

What's more, they don't know their value to you. They may feel entitled to a free product because that's how you've positioned your SaaS offering, or to free breadsticks because that's how you've priced your buffet. You know their value to your business—they don't. To each unhappy user he or she is, they're the most important person in the world. And he or she has been wronged or celebrated.

Finally, customers aren't aware of the constraints and nuances of their problems. It's easy to complain about US television programming not being available overseas; it's unlikely that those complaining are aware of the intricacies of foreign currency exchange, censorship, and copyright licensing. They want their problem solved, but they have little insight into how to solve it the *right* way.

Laura Klein is a user experience (UX) professional and consultant, as well as the author of *UX for Lean Startups* (forthcoming from O'Reilly), part of the Lean Series along with this book. She writes a great blog called Users Know. You should read her post, "Why Your Customer Feedback is Useless," in its entirety.

[\[59\]](#)

To improve how you interpret feedback, Laura has three suggestions:

- Plan tests ahead of time, and know what you want to learn before you get started. "A big reason that feedback is hard to interpret is because there's just too much of it, and it's not well organized or about a particular topic," says Laura. "If you know exactly what you're gathering feedback on, and you're

disciplined about the methods you use to gather it, it becomes very simple to interpret the responses.”

- Don’t talk to just anybody. “You should group feedback from similar personas,” says Laura. “For example, if I ask a Formula 1 driver and my mom about how they feel about their cars, I’m going to get inconsistent responses.” Balancing feedback like that is very difficult because it’s from such different types of people. “Figure out who your customers are and focus your research on a particular type of person.”
- Review results quickly as you collect data. “Don’t leave it all until the end,” Laura notes. “If you talk to five people for an hour each over the course of a few days, it can be really hard to remember what the first person said.” Laura recommends having someone else in each session, so that you can debrief with that person after and pull out the top takeaways from the session.

The reality is, users will always complain. That’s just the way it goes. Even if people are using your product, you have good engagement metrics, and your product is sticky, they’re still going to complain. Listen to their complaints, and try to get to the root of the issue as quickly as possible without overreacting.

The Minimum Viable Vision

The *minimum viable vision* is a term coined by entrepreneur and Year One Labs partner Raymond Luk. He says, “If you’re trying to build a great company and get others involved, it’s not enough to find an MVP—you need an MVV, too.”

A minimum viable vision (MVV) is one that captivates. It scales. It has potential. It’s audacious and compelling. As a founder, you have to hold that huge, hairy, world-changing vision in one hand, and the practical, pragmatic, seat-of-the-pants reality in the other. The MVV you need in order to get funding demands a convincing explanation of how you can become a dominant, disruptive player in your market.

Here are some signs that suggest you’ve got the makings of an MVV:

- ***You’re building a platform.*** If you’re creating an environment in which other things can be created, this is a good sign. Google Maps was just one of the

many mapping tools available, alongside MapQuest and others, but Google made it easy to embed and annotate those maps, leading to thousands of mashups and clever uses. It quickly became the de facto platform for entry-level geographic information systems (GIS), and all those annotations made its maps even more useful.

- ***You have recurring ways to make money.*** It's one thing to take money from someone once, but if you can convince that person to pay every month as well, you're onto something. Just look at Blizzard's revenues from *World of Warcraft*: purchase of the paid desktop client is a fraction of the money the company makes compared to revenue from \$14.95 per month subscriber fees.
- ***You've got naturally tiered pricing.*** If you can find ways for customers to self-upsell, as companies like 37Signals, Wufoo, and FreshBooks have done, then you can hook your users on basic features and tempt them with an upgrade path that adds functionality as they need it. This means you'll not only add revenue from new users, but from existing ones, too.
- ***You're tied to a disruptive change.*** If you're part of a growing trend—people sharing information, mobile devices, cloud computing—then you've got a better chance of growth. A rising tide floats all boats, and a rising tech sector floats all valuations and exits.
- ***Adopters automatically become advocates.*** Just look at the classic example of online marketing—Hotmail. A simple message appended to every email invited the recipient to switch to Hotmail. The result was an exponential growth rate and a huge exit for the founders.^[60] An expense management system like Expensify makes it as easy as possible to add others to the approval workflow, because this is a vector for inherent virality.
- ***You can create a bidding war.*** If you've got a solution that several industry giants will want, you're in a great place. While big companies can build anything given enough time, they'll buy you if you're stealing their sales or if your product helps them sell dramatically more easily. Beverage giants like Pepsico, Cadbury-Schweppes, and Coca-Cola regularly buy out promising incumbents, like Odwalla, Tropicana, Minute Maid, RC Cola, and others, knowing they can make back their investment easily through their existing

supply chains.

- ***You're riding an environmental change.*** We don't mean the Green movement here. In strategic marketing, environmental forces include everything you're subject to in your business ecosystem, such as government-mandated privacy laws or anti-pollution regulations. If you're building something that everyone will be forced to adopt (such as a product that complies with soon-to-be-signed health or payment privacy legislation), you've got a promising exit and a chance to take over the sector.
- ***You've got a sustainable unfair advantage.*** There's nothing investors like more than unfairness. If you can maintain an unfair advantage—lower costs, better market attention, partners, proprietary formulae, and so on—then you can scale your business to a degree where it's interesting to investors. But be careful: outside of government-mandated monopolies, few advantages are truly sustainable in the long term.
- ***Your marginal costs trend to zero.*** If as you add users your incremental costs go down—so that the n th customer costs almost nothing to add—that's a great place to be. You're enjoying healthy economies of scale. For example, an antivirus company has fixed costs of software development and research that must be amortized across all customers, but the addition of one more client adds only a vanishingly small cost to this total. Businesses that can grow revenues while incremental costs stay still or decline have the potential to grow massively overnight.
- ***There are inherent network effects in the model.*** The phone system is the classic example of a business with a network effect: the more people who use it, the more useful it becomes. Network-effect businesses are wonderful, but they often have a two-edged sword: it's great when you have 10 million users, but you may be deluding yourself about how easily users will adopt the product or service, and it's hard to test the basic value with a small market at first. You need a plan for getting to the point where the network effects kick in and become obvious.
- ***You have several ways to monetize.*** It's unlikely that any one payment model will work, but if you can find several ways to make money from a business—

one obvious one, and several incidental ones—then you can diversify your revenue streams and iterate more easily, improving your chances of success. Quick note: AdWords and selling your analytical data probably aren't enough.

- ***You make money when your customers make money.*** Humans are, at their most basic, motivated by two things: love and fear. Love means things like costs and risks, and if you reduce risks or cut costs, that's nice—but it's not compelling. Customers will often rationalize away the risk and pocket the savings. But if you make money from revenues, then the customer will likely split the winnings with you. Products that boost revenues are easier for people to believe in—just look at lotteries and get-rich-quick schemes versus savings plans and life insurance. Eventbrite and Kickstarter know this.
- ***An ecosystem will form around you.*** This is similar to the platform model. Salesforce and Photoshop are good examples of this: Salesforce's App Exchange has thousands of third-party applications that make the CRM (customer relationship management) provider more useful and customizable, and Photoshop's plug-in model added features to the application far more quickly than if Adobe had coded them all itself.

In the end, you have to be audacious. You need to understand how your company can become a Big Idea, something that's truly new, and either widely appealing to a broad market or a must-have for a well-heeled niche.

The Problem-Solution Canvas

At Year One Labs, we developed a tool called the *Problem-Solution Canvas* to help our startups maintain discipline and focus on a weekly basis. It's inspired by Ash Maurya's Lean Canvas, but focused on the day-to-day operations of a startup. We used it to home in on the key one to three problems the startups were facing. It allowed us all to agree on those problems and prioritize them.

It was fairly common for founders to incorrectly prioritize the key issues at hand. It's not surprising; startup founders are juggling a ton at once, wearing hats stacked to the sky like crazed circus performers, and as we well know, they're a bunch of liars (but we love 'em just the same!). As mentors and advisors, we

knew that a big part of our job—where we could provide significant value because of our detachment—was to guide entrepreneurs back to what was most important.

The Problem-Solution Canvas is a two-page document. Like a Lean Canvas it's divided into a few boxes. On a weekly basis we'd ask founders to prepare a Problem-Solution Canvas and present it. The canvas became the focal point for our status meetings, and it was extremely helpful for keeping those meetings productive.

Figure 16-3 shows the first page of the template.

THE GOAL IS TO LEARN	
CURRENT STATUS <ul style="list-style-type: none">- List key metrics you're tracking, where they're at, and compare with last few weeks- How are things trending?	LAST WEEK'S LESSON LEARNED (AND ACCOMPLISHMENTS) <ul style="list-style-type: none">- What did you learn last week?- What was accomplished?- On track: YES / NO
TOP PROBLEMS <ul style="list-style-type: none">- List and describe the top problems- Prioritize them	

Figure 16-3. If you filled in this page every week, what would you learn?

The first thing you'll notice is the title: *The Goal Is to Learn*. This is important, because it reminded the entrepreneurs about what they were setting out to do. It wasn't about building "stuff." It wasn't about adding features. It wasn't about getting PR, or anything else. Learning was the measure of success.

Next, founders would fill in a brief update on their current status, focusing on the key metrics (qualitative and/or quantitative) that they were tracking. Notice how small this box is compared to the others.

The Lessons Learned box is a quick bulleted summary of key learning. The title says "and Accomplishments" because we wanted to give entrepreneurs a place to brag—at least a little bit. Not surprisingly, they'd include some vanity metrics

to brag—at least a little bit. Not surprisingly, they do include some vanity metrics in here and we wouldn't spend a lot of time on them. The “On track: Yes/No” benchmark is designed as a test of intellectual honesty. Can entrepreneurs really come clean on what's going on, good and bad? If so, we could be much more valuable.

Finally, we asked entrepreneurs to list the top problems they were facing at that moment. At most they would include three problems prioritized in order of importance. This section of the Problem-Solution Canvas often elicited the most debate, but it was always healthy and critical for resetting everyone's goals and expectations.

With the problems now well understood, along with the startup's current status, we'd move to the second page of the canvas, shown in **Figure 16-4**.

Problem #1: [put name of it here]	
HYPOTHESIZED SOLUTIONS <ul style="list-style-type: none">- List possible solutions that you'll start working on in the next week. Rank them.- Why do you believe each solution will help solve or completely solve the problem?	METRICS / PROOF + GOALS <ul style="list-style-type: none">- List metrics you'll use to measure whether or not the solutions (to the left) are doing what you expected (solving the problem)- List proof (qualitative) you'll use as well- Define goals for the metric
Problem #2: [put name of it here]	
HYPOTHESIZED SOLUTIONS <ul style="list-style-type: none">- List possible solutions that you'll start working on in the next week. Rank them.- Why do you believe each solution will help solve or completely solve the problem?	METRICS / PROOF + GOALS <ul style="list-style-type: none">- List metrics you'll use to measure whether or not the solutions (to the left) are doing what you expected (solving the problem)- List proof (qualitative) you'll use as well- Define goals for the metric

Figure 16-4. We've all got problems—but can you pick just three?

In this section, the founders re-list the problems and include hypothesized solutions. These solutions are hypothesized because we don't know if they'll work. These are experiments that the founders will run in the next week. We always asked them to define the metrics they'd use to measure success (or

failure) and draw a line in the sand. If engagement was the most important problem, they had to include possible solutions they'd experiment with to increase engagement, define the metric (e.g., % daily active users), and set a target. *What's the problem, how do you propose to fix it, and how will you know if you succeeded?* That's the core of the Problem-Solution Canvas.

For us (as mentors and advisors), it was an extremely valuable exercise. The Problem-Solution Canvas is also useful for internal decision making. It sits a level below the Lean Canvas, focusing on very specific details in a very specific time period (one to two weeks).

VNN Uses the Problem-Solution Canvas to Solve Business Problems

Varsity News Network (VNN) is an early-stage startup based in Michigan. Ben met one of the founders there, Ryan Vaughn, when speaking at a conference in 2012. The company's platform makes it easy for athletic directors to manage social communication, creating hyper-local media coverage about athletics at their high schools. The goal is to leverage that awareness creation into ongoing financial and emotional support for high school sports.

Ryan was introduced to the Problem-Solution Canvas and started using it immediately with his board of directors. "We had just raised financing and had to solve a number of key business problems very quickly," said Ryan. "We used the Problem-Solution Canvas to get all our board members on the same page, focused on what we had to do in order to move forward."

VNN followed a Lean process, particularly in the beginning of the company in order to determine its value proposition and how that tied into producing content about high school sports. The company remains Lean today, testing and iterating each new feature or initiative it launches, measuring effectiveness and value creation.

Still, Ryan was concerned that his board wouldn't embrace the Problem-Solution Canvas. He said, "The Lean Startup process has not been widely adopted in the Midwest yet, but our board had been exposed to the methodology, which helped speed up our initial progress with the canvas."

VNN used the canvas for a few months, during a critical time of problem

solving. The result was that everyone involved stayed focused on the major tasks at hand. Through the Problem-Solution Canvas, VNN validated a number of its core assumptions and designed a scalable growth model involving direct sales. This allowed it to prove enough of its business to start generating revenue and plan for a second round of financing.

Figure 16-5 and Figure 16-6 show an example of one of its canvases.

MAY PROBLEM/SOLUTION DASHBOARD	
METRICS	LAST MONTH'S LESSON LEARNED (AND ACCOMPLISHMENTS)
<ul style="list-style-type: none">- Schools sold: 1- Last month: 3- Total: 34- Ad sales/school: \$4,750- Ad sales/rep/mo: \$6,150- Traffic/school: 1931.9 visits- +200 visits over last 3 months (new theme)	<ul style="list-style-type: none">- What did you learn last month?<ul style="list-style-type: none">- One sales rep can sell \$10k+ in one month- A market exists for individual sport websites- Schedules > photos > articles- What was accomplished?<ul style="list-style-type: none">- Photogs sponsored for right to submit pics- Secured second test market in Indianapolis- On track: YES
TOP PROBLEMS	
<ul style="list-style-type: none">- 1. We still don't know what one full-time rep can sell<ul style="list-style-type: none">- We believe \$10k in ads and 2 schools, based on limited data- 1 part-time rep sold 5+ schools in multiple months- 1 part-time rep sold \$10,000 in ads two months in a row- 2. We don't know the market for sport-specific sites<ul style="list-style-type: none">- Coaches will pay \$20–30/month. How should we sell/support these?- 3. We don't yet show enough value to advertisers<ul style="list-style-type: none">- We will need a 50%+ renewal rate year-over-year, current rate = 50%	






Figure 16-5. VNN spends some time on introspection

Problem #1: We still don't know what 1 full-time rep can sell in a month

HYPOTHESIZED SOLUTIONS	METRICS / PROOF + GOALS
<ul style="list-style-type: none">- 1. Hire full-time rep in Ann Arbor to sell both schools and ads in East Michigan<ul style="list-style-type: none">- Shows what average rep can do- 2. Contract with two guys out of Indianapolis to start new market<ul style="list-style-type: none">- Shows what elite-rep can do	<ul style="list-style-type: none">- Metrics: School and ad sales/rep- Ann Arbor: Will have sold \$8,500 and 3 schools by end-July- Indianapolis: Will have sold \$7,500 and 4 schools by end-July

Problem #2: We don't know the market for sport-specific websites

HYPOTHESIZED SOLUTIONS	METRICS / PROOF + GOALS
<ul style="list-style-type: none">- 1. Interview coaches in/out of Michigan<ul style="list-style-type: none">- Big questions are if there is demand, pricing, and features- 2. If market, build and sell MVP to coaches<ul style="list-style-type: none">- This is the ultimate test of market- Question is <i>how</i> best to sell it	<ul style="list-style-type: none">- Metrics: Interview responses and sales- 1. Interviews with coaches<ul style="list-style-type: none">- Stated interest and payment amount- Pre-purchase orders



Figure 16-6. Knowing how much you can sell, and the size of the market, matters a lot

Summary

- Having raised funding, VNN used the Problem-Solution Canvas to communicate with its board of directors in an effective manner.
- The canvas helped the company iterate to revenue and position itself for additional financing.

Analytics Lessons Learned

Never underestimate the power of getting everyone on the same page—literally. A single sheet of consistent information that forces all stakeholders to be succinct and to agree really helps clarify and define a problem, particularly in a fast-changing environment.

A Summary of the Stickiness Stage

- Your goal is to prove that you've solved a problem in a way that keeps people coming back.
- The key at this stage is engagement, which is measured by the time spent interacting with you, the rate at which people return, and so on. You might track revenue or virality, but they aren't your focus yet.
- Even though you're building the minimal product, your vision should still be big enough to inspire customers, employees, and investors—and there has to be a credible way to get from the current proof to the future vision.
- Don't step on the gas until you've proven that people will do what you want reliably. Otherwise, you're spending money and time attracting users who will leave immediately.
- Rely on cohort analysis to measure the impact of your continuous improvements as you optimize the stickiness of your product.

When your engagement numbers are healthy and churn is relatively low, it's time to focus on growing your user base. Don't run out and buy ads immediately, though. First, you need to leverage the best, most convincing campaign platform you have—your current users. It's time to go viral.

Exercise #1: Should You Move to the Next Stage?

1. Are people using the product as expected?
 - If they are, move to the next step.
 - If they aren't, are they still getting enough value out of it, but using it differently? Or is the value not there?
2. Define an active user. What percentage of your users/customers is active? Write this down. Could this be higher? What can you do to improve engagement?

3. Evaluate your feature roadmap against our seven questions to ask before building more features. Does this change the priorities of feature development?
4. Evaluate the complaints you're getting from users. How does this impact feature development going forward?

Exercise #2: Have You Identified Your Biggest Problems?

Create a Problem-Solution Canvas. This should take no more than 15–20 minutes. Share your canvas with others (investors, advisors, employees) and ask yourself if it really addresses the key concerns you're facing today.

[57] One exception to this rule is a business that requires a critical mass of activity to be useful. If your service is engaging only when it has, say, 1,000 property listings, or 10,000 prospective mates, or cars less than three minutes away, then you'll need to artificially seed it somehow before you can focus on testing stickiness. This is a common problem for two-sided marketplaces.

[58] <http://www.amazon.com/Art-Focused-Conversation-Access-Workplace/dp/0865714169>

[59] <http://usersknow.blogspot.ca/2010/03/why-your-customer-feedback-is-useless.html>

[60] <http://www.menlovc.com/portfolio/hotmail>