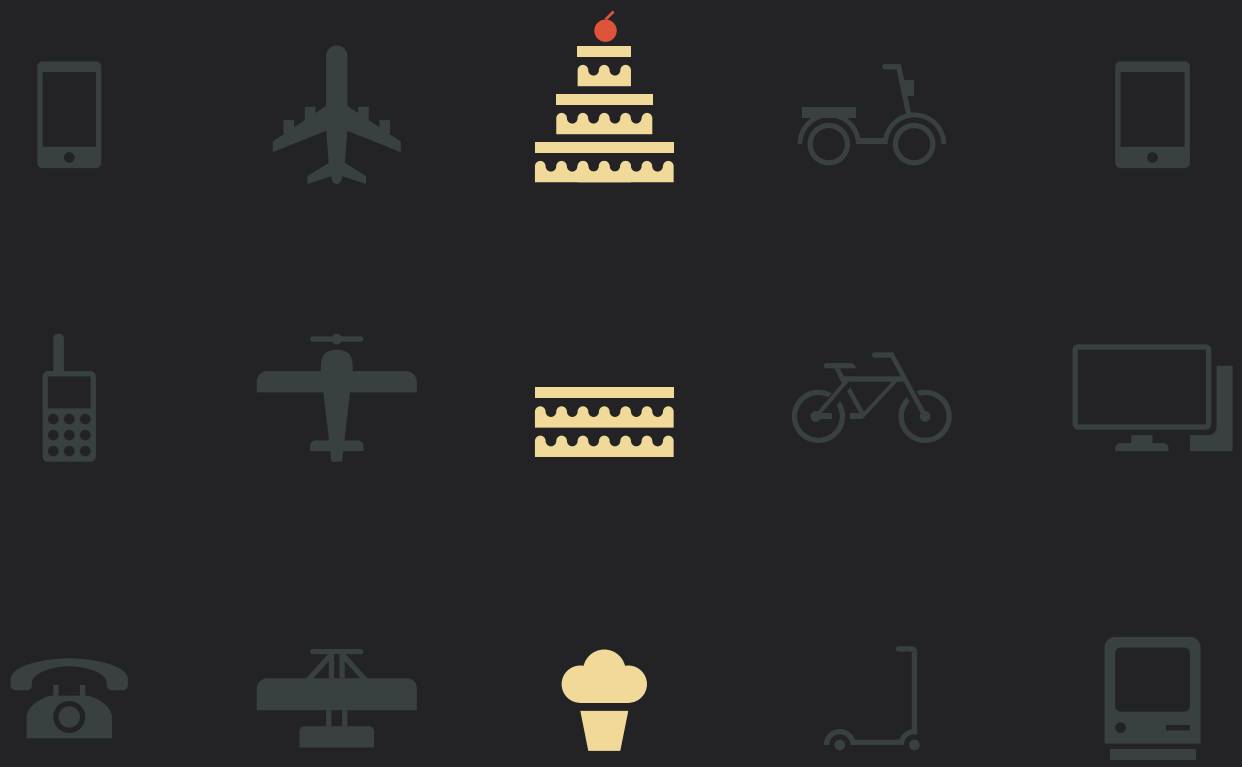
UXPin

THE GUIDE TO

MINIMUM VIABLE PRODUCTS

A Master Collection of Frameworks, Expert Opinions, and Examples



INDEX

CHAPTER ONE

INTRODUCTION

A quick note from the author

Minimum viable products are frequently misunderstood and misused. Because most business are strapped for resources, it's a common mistake to focus only on the "minimum" part to get something out the door quickly.

But an MVP is much more than just a minimum product. It is perfection by subtraction, the best bang for your buck, a serious reality check — "Hello, World!". An MVP trims all the fat and leaves just the essence of your value to customers with the resources.

In this book, we'll share a wide breadth of expert commentary, theories, practices, and real-life examples of MVP success and failure. To name a few, we've included advice from entrepreneurs like ***Steve Blank, Eric Ries, Guy Kawasaki, Ash Maurya, Andrew Chen, Cindy Alvarez, Rand Fishkin, David Aycan, Joel Gascoigne, Josh Puckett, Brandon Schauer, Chrys Bader, Neil Patel, Nick Swinmurn***, and more. We'll discuss basic concepts like the different types of MVPs and how to test hypotheses with MVPs. For more experienced readers, we've also laid out how to apply MVP thinking in a Lean and Agile environment, how to balance UX with Lean development, and even Spotify's internal design process. Our hope is that it will help you better understand how to strike the perfect balance between resource minimalism, business viability, and product quality in your next MVP.

When you think about it, testing an MVP is probably the most important step to success for companies. We'll look at how highly successful companies like ***Twitter, Zynga, Foursquare, Dropbox, Zappos, Groupon, Oculus VR, Airbnb, Buffer, Pebble*** among others built the right MVP for the right reasons to help them refine their business idea and get people buzzing about their products. We've also included our own story and outlined how you can use UXPin to help prototype your own MVP.

We'd love your thoughts on what we've written. And feel free to include anyone else in the discussion by sharing this e-book.

For the love of minimum viable products,

Chris Bank
(co-written by Jerry Cao & Waleed Zuberi)

Chris Bank is the growth lead @UXPin. He also led growth @Lettuce (acquired by Intuit),@MyFit (acquired by Naviance), and his own startup @Epostmarks (USPS strategic partner), and launched @Kaggle in the B2B tech vertical. In his downtime, he rock climbs, motorcycles, designs apps, travels, and reads. Visit my website and Follow me on Twitter.

Jerry Cao is a content strategist at UXPin where he gets to put his overly active imagination to paper every day. In a past life, he developed content strategies for clients at Brafton and worked in traditional advertising at DDB San Francisco. In his spare time he enjoys playing electric guitar, watching foreign horror films, and expanding his knowledge of random facts. Follow me on Twitter.

Waleed Zuberi is passionate about creating better user experiences through thoughtful design. When he's not writing or pushing pixels on the web, he enjoys biking, playing cricket and binge-watching TV. Visit his website and follow him on Twitter.

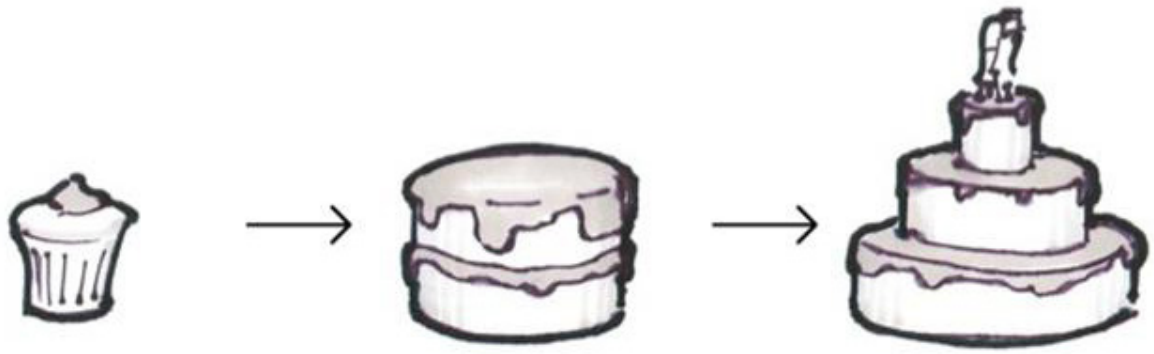
CHAPTER TWO

MINIMUM VIABLE PRODUCTS -DEFINED BY THE EXPERTS

How the top product minds in the world think about MVPs

Today, lean startups and tech titans alike are increasingly using the minimum viable product (MVP) as a starting point for building successful software products.

By focusing on an integral set of key features and core functionality for product development, firms can efficiently establish a definitive core to form the basis out of which the rest of the product can evolve. If they can't get this right, they risk ending up with a product that SUX — an offering with a "Sh***y User Experience."



Source: Lean Heroes

WHAT'S AN MVP?

Startups and tech titans alike use varying measures for defining what goes into an MVP, and many are still slightly misguided.

A common misconception is that an MVP consists of the minimum set of features deemed necessary for a working software product, with the goal of bringing it to market quickly. This misses the mark on several levels, most notably in ***the over-emphasis on speedy delivery and time to market, as opposed to focusing on customer and market acceptance.*** Indeed, rapid development is of essence, but only to the extent that learning and research objectives can be obtained quickly.

As defined by Wikipedia, *"The minimum viable product (MVP) is a strategy used for fast and quantitative market testing of a product or product feature. The term was coined by Frank Robinson and popularized by Eric Ries for web applications."* This definition is narrow — particularly, it's too quantitative and product-focused — according to many experts. However, some of the noted purposes of an MVP below begin to open up a more significant discussion:

- Be able to test a product hypothesis with minimal resources
- Accelerate learning
- Reduce wasted engineering hours
- Get the product to early customers as soon as possible

Let's look at what the experts have to have to say.

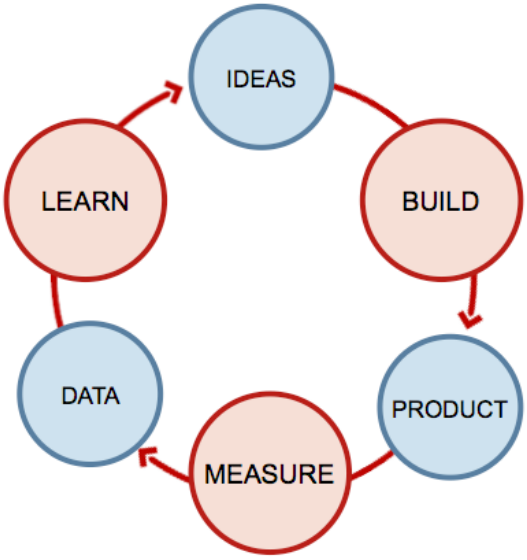
EXPERTS' TAKES ON MVPS

It's important for you and your team to form your own opinion about what an MVP means to you, but hopefully the viewpoints of some notable executives below help you flesh that out.

Sources: Eric, Ash, Nick, Steve, Rand, Cindy and Marcin

Eric Ries, cofounder/CTO of IMVU and MVP proponent, defines an MVP as a version of a new product that allows for the most learning possible for the least amount of effort. That is to say, an MVP allows for testing actual usage scenarios with customers. To this end, expensive market research and subsequent product development is eschewed; instead, a rapidly-built product with a minimum set of features is deployed to test assumptions about customer requirements.

You've probably already heard this definition enough times to make you scream. So let's round this off with other helpful perspectives.

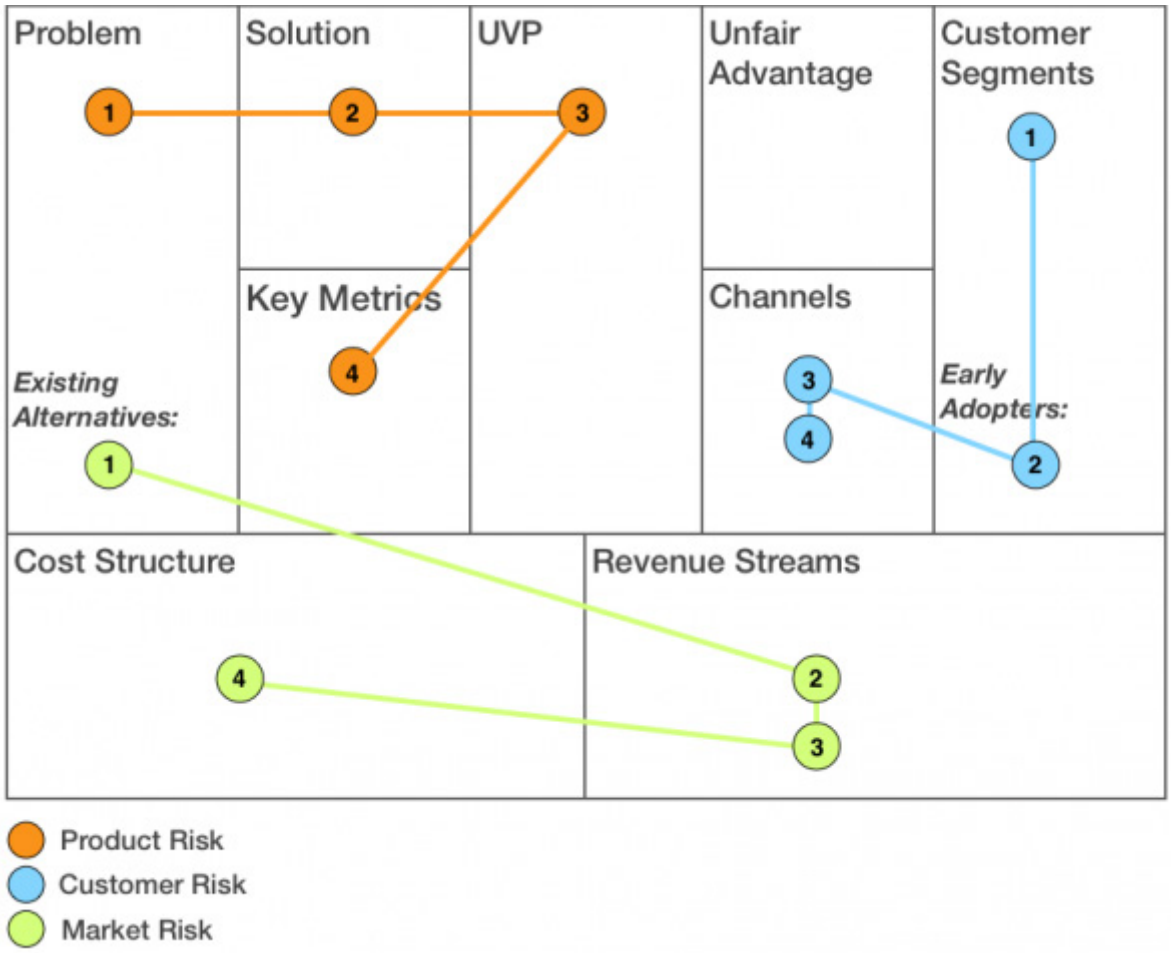


Source: Build Measure Learn

Ash Maurya, CEO of P2P sharing site Cloudfire and author of *Running Lean - Helping Entrepreneurs Succeed*, recounts his experiences in building an MVP for his startup. After identifying a target group of users, he then proceeded to identify the three main issues they experienced with current solutions on the market. He was then able to build a solution to minimally address those issues, and drive early adopters to sign up via the product's landing page. Fortunately for Maurya, the process was simplified through leveraging key functionality from a previous product. This allowed him to dramatically cut the time and effort it took him to validate his assumptions about the potential user base for his product. Reflecting on his experience, Ash now emphasizes the importance of capturing customer value with any MVP. It's critical to get the product right, so make sure you have a problem worth solving. Using the Lean Canvas framework below, he highlights the 4 critical steps to nailing the product in your MVP:

- First make sure you have a problem worth solving.
- Then define the smallest possible solution (MVP).
- Build and validate your MVP at small scale (demonstrate UVP).

- Then verify it at large scale.
- To understand market and customer risks of an MVP, see his post [The 10x Product Launch](#).

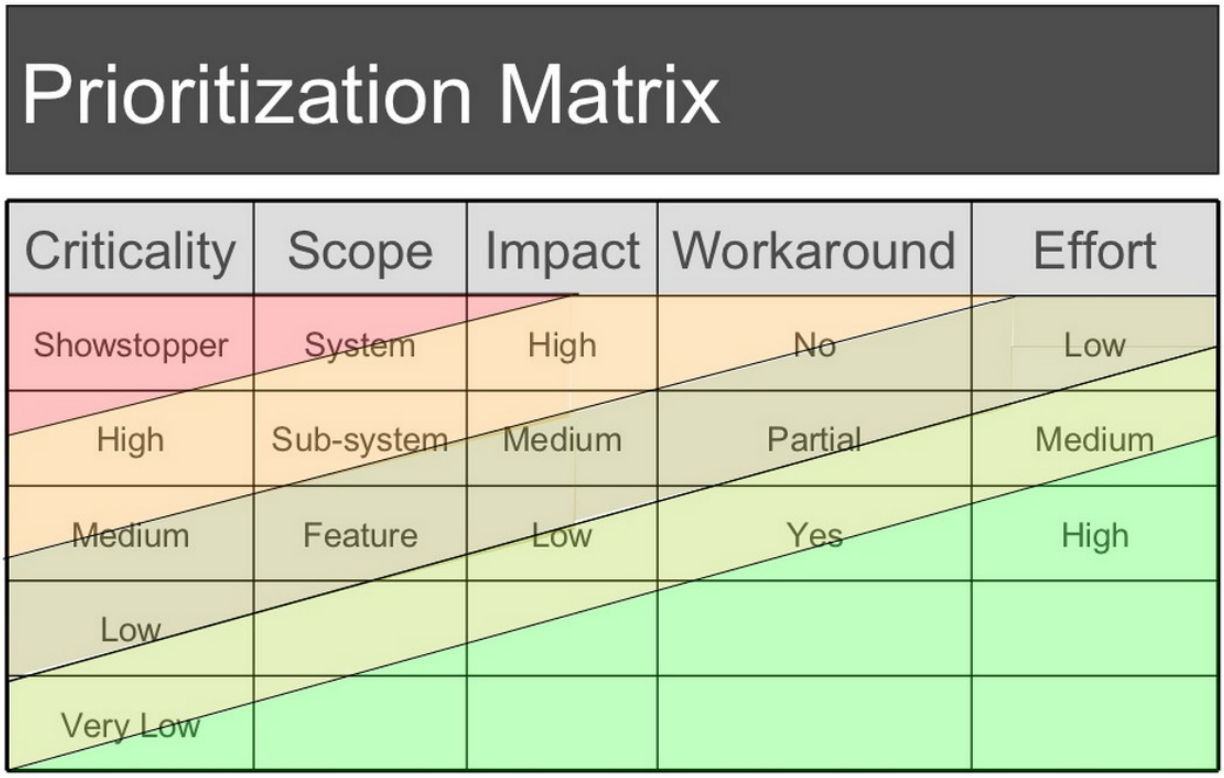


Source: [The 10x Product Launch](#)

Marcin Treder, CEO and Co-Founder of UXPin, went through a very similar experience although his company's existing product was a paper prototyping notepad while the current solution is a web-based wireframing and prototyping application. "Clearly, the paper products were cheaper to make initially," he states, "But we honestly had no idea that our next version of the product would be technical — we were a few designers just trying to help our peers become better designers."

"An MVP requires minimum development effort to create maximum value." Share Quote

But he quickly realized how fragmented the existing solutions were and how much people complained about them.According to Treder, "An MVP isn't the quickest or the most perfect product. Rather, it is a product with minimum development effort that creates maximum value." He admits that his first product didn't provide the maximum value given current alternatives today. But, today, UXPin is one of the leading wireframing and prototyping applications on the market. So he clearly made the transition.

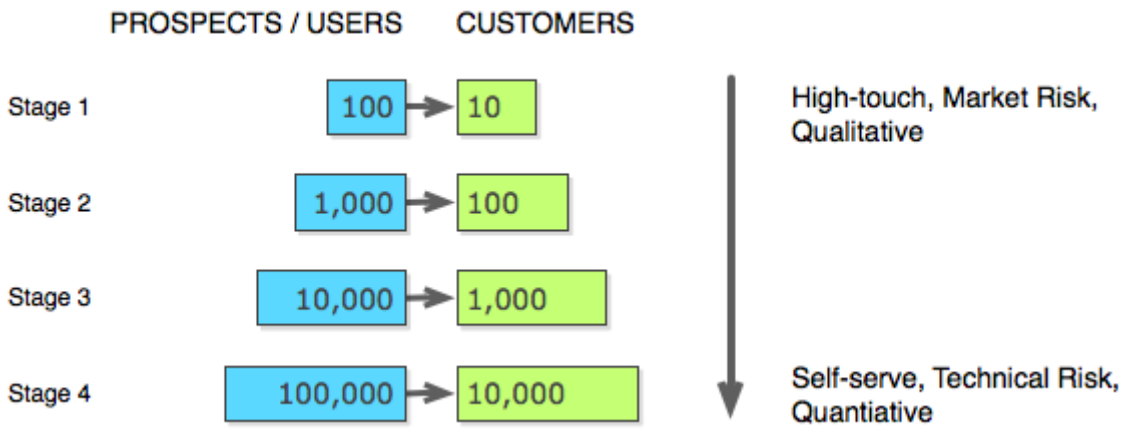


Source: Practical Product Management for New Product Managers

To get the most value out of development efforts, you can use the above matrix to systematically prioritize features. Of course, not all lean startups will have the luxury of having a pre-existing product to massage into MVP form. In many cases, this is barely necessary.

Nick Swinmurn, Zappos co-founder, experienced this first-hand**.** In an

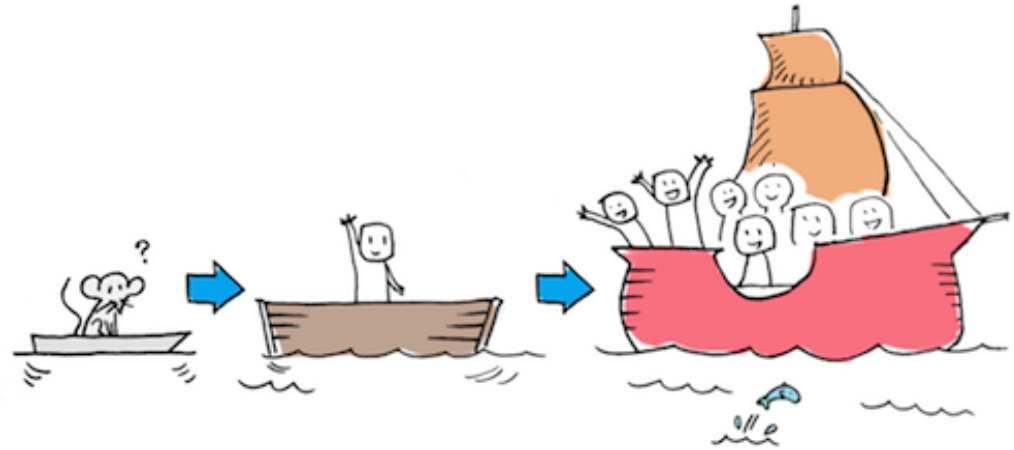
extreme case of MVP leanness and agility, the e-commerce stalwart's humble beginnings started with Swinmurn photographing shoes at a local retailer. He then posted the photos online and, for each online order, he would return to the retailer and buy the necessary items. In this sense, the primary objective of an MVP is to eliminate business uncertainty to the greatest extent possible. He didn't have a product and acted as his own customer in the initial stages.



Source: The 10x Product Launch

And it's far more common for companies to sell and market vaporware (products that don't yet exist or barely exist) — especially in the startup world.

Cindy Alvarez, UX for Yammer and previously Product at Kissmetrics, echoes that a common mistake people make is assuming an MVP needs to be a product. According to her, the goal of an MVP is to maximize learning while minimizing risk and investment and, therefore, a product should not be the only means to that end. By thinking so narrowly about MVPs, she has seen many people start by thinking about the "final" product and trying to cut features instead of doing anything scientific. To avoid this pitfall, one of her rules of thumb is following the Cupcake Model whereby you think of one complete experience.



Source: 7 Ways to Test MVPs

She also provides some practical advice in executing on MVPs within teams. She claims there are two important problems that must be addressed head-on internally:

1. **Set expectations appropriately** — No one wants to build something crappy and feel that they will never get the chance to make it better. Emphasize that the purpose of MVPs is to make sure your team isn't wasting its time on worthless products and features and that your team can actually improve the product using validated testing.

2. **Set your MVP target customer appropriately** — Be careful about building for a mainstream audience. If you do, they may tell you it sucks and you'll get the wrong signals about what you're building. Instead, find customers with an identified early pain and show them your early, sloppy MVP that is supposed to solve their problem.

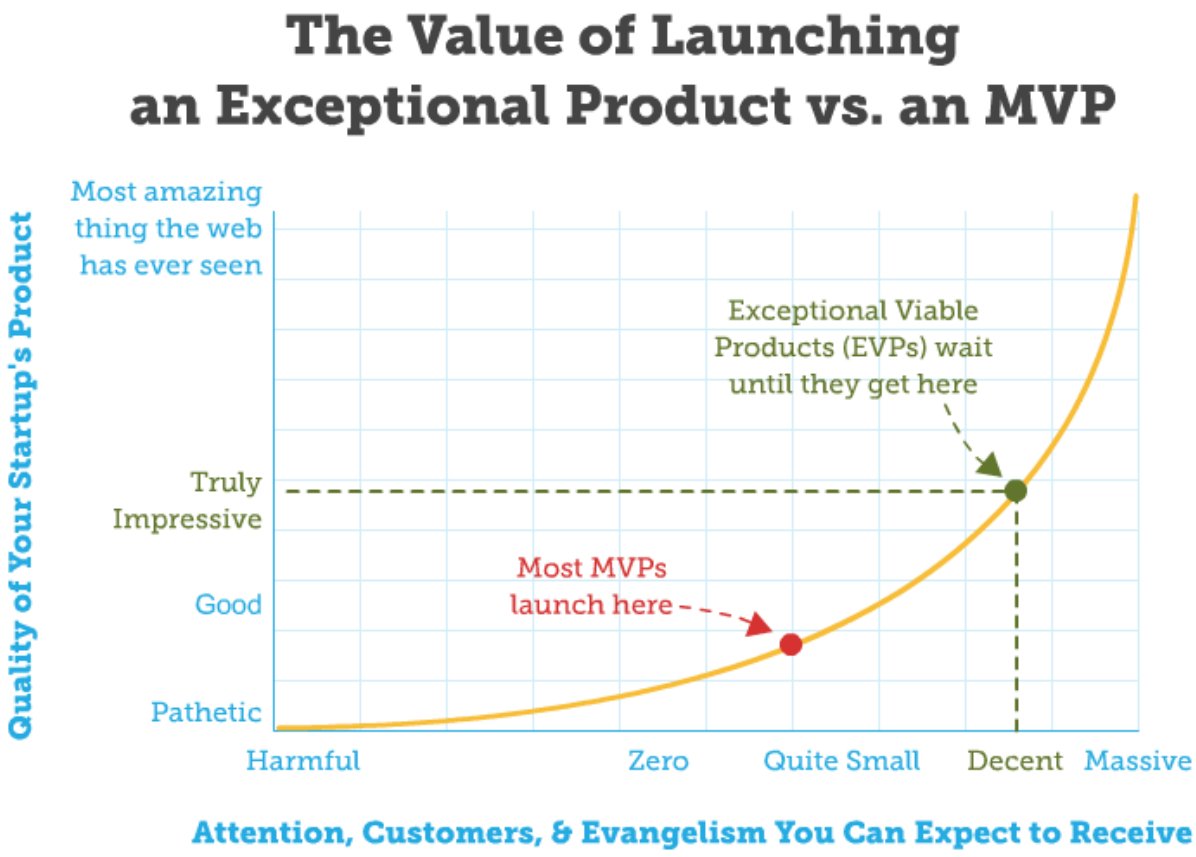
Steve Blank, a serial-entrepreneur and author / lecturer on MVPs, asserts that the Customer Development and Lean Startup methodology under-emphasizes the importance of selling a vision to visionaries — not everyone — while delivering a minimum feature set. From his observations, many people easily comprehend how to build a minimal product with few features (the Minimum

Feature Set), but they fail to acknowledge that most people won't like their MVP — how many people do you know will brag about a minimal product? Instead, companies should be building early adoption and evangelism while selling a vision about how the world will work, and be so much better, with the product being built a few years out based on a minimal product that you can play around with today.

Source: Silicon Africa, 5 Characteristics of Earlyvangelists

Rand Fishkin, co-founder of Moz, seems to be more of a showman than the rest. From his perspective, first impressions matter — a lot. It is for this reason that he encourages others to take their MVP one step further toward being an EVP (an Exceptional, Viable Product). He claims to have seen a lot of MVPs launch that hardly produce significant value, and strongly believes it's highly problematic. After all, there's only so many times you can re-launch a product. In practice, Rand suggests making your MVPs in-house and dogfooding them internally and with a few customers. Gather feedback and iterate until the first internal and external users find that "A Ha" moment, then release it to the wild as an EVP. This

may take an extra 30-90+ days to reach this point but, in his opinion, it's well worth the wait.



Source: [Moz, "7 unlikely recommendations for startups & entrepreneurs"]

MVPS ARE AN IMPORTANT MEANS TO AN END

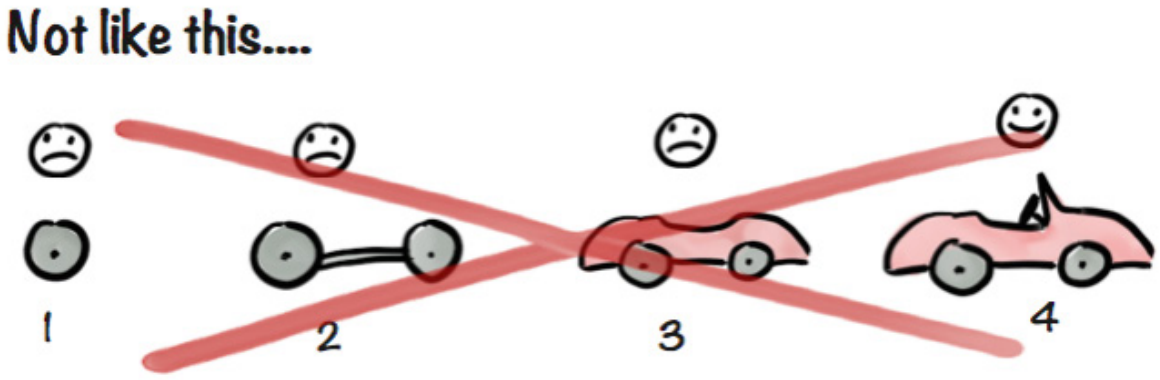
There are many ways to skin a cat, but even more ways to deliver an MVP. Although each expert has their own twist on what an MVP means to them and the golden rules they follow to make sure they don't get caught in the weeds, the underlying message is the same: MVPs are a means to an end product or product improvement, not the end product itself. Make sure you don't lose sight of that.

CHAPTER TWO

M IN MVP: EXPERT TAKES ON PRODUCT MINIMALISM

How to think about constraints when building your next product

Applying minimalism is perhaps the most difficult but important part of the MVP process. After all, your interpretation determines the strategy behind selecting the right features and technical resources. Use the right amount of minimalism and you create an elegant gem that is low-cost and high-learning. Go overboard with trimming your features and you end up with a shoddy prototype that doesn't just fail at answering your hypotheses but could embarrass the brand.



Source: Stop Overthinking... Just Stop

So how do you build a stripped-down product that is affordable and appealing? Start by understanding the difference between these 2 questions:

> **Question A:** How can we build the simplest technically feasible product?
> **Question B:** How can we build the simplest product to resonate with early adopters?

The first approach prioritizes deliverability and can result in the mistake of using a set of tires to test the concept of a car. The second approach focuses on the core value of the product, a principle that is much more helpful towards uncovering the learnings you need.

Read on to hear expert advice on MVPs and how you can use minimalism to develop a high-quality, focused experiment for your most important users.

SEIZE YOUR UNFAIR ADVANTAGE QUICKLY

When designing the UXPin MVP for the US market, we interviewed influencers to gather feedback on what features and functionality were missing in today's UX tools. We bounced around ideas on product strategy and even sketched wire-frames on napkins. We sought their advice to better understand the early adopter mindset and prevent ourselves from releasing a "minimum product" that would fail to convey our vision.

Former Apple chief evangelist, Guy Kawasaki, asserts in his MVP philosophy that the MVP does not need to be perfect but it does need to be revolutionary. The goal of minimalism then is to reduce engineering waste by only incorporating enough features that embody your unfair advantage to capture the interest of early adopters. These "earlyvangelists" (coined by serial entrepreneur Steve Blank) can make or break your idea, so focus your MVP on the soul of your product to best learn from these force-multipliers.

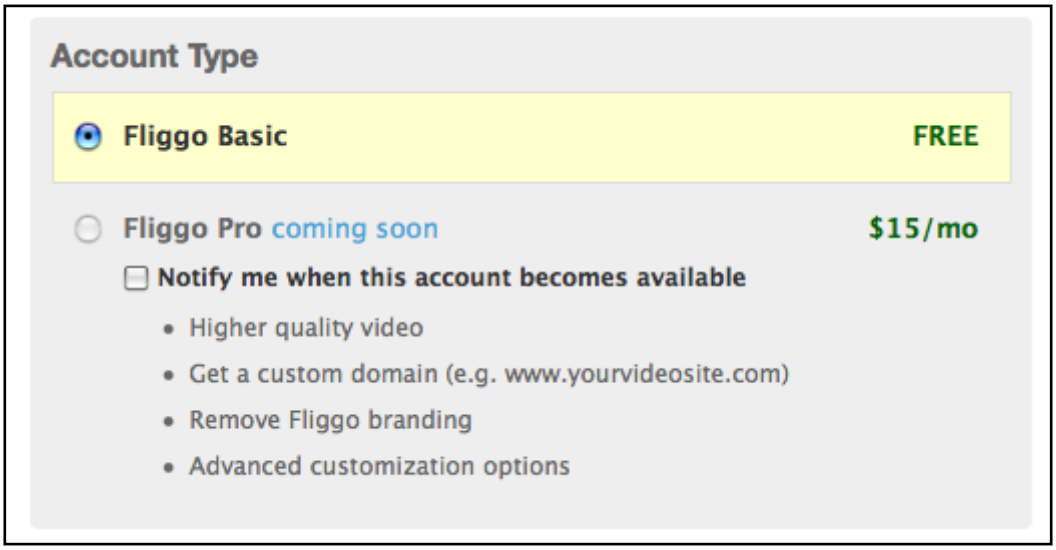
According to **Matchist.com cofounder, Stella Fayman,** in a piece on MVPs for KISSMetrics, balance these 3 questions to stay on track:

- Are resources dedicated to simplifying the MVP?
- Are the assumptions focused just on the core value?
- Is my timeline as lean as possible?

Focusing on question #2 above is extremely important for anchoring your minimalistic approach in customer reality. As Fayman suggests, start by mapping your functionalities to assumptions and add more layers of functionality into your MVP only as you prove each assumption. This ensures that no matter the depth of your feature set, you're always executing them properly for a consistent early

adopter experience.

It can be hard to understand minimalism abstractly, so let's look at a real-world example below.



Source: Sell It Before You Build It

At Fliggo (merged with Vidly), Chrys Bader, currently the co-founder and Chief Product Officer at Secret, wanted to test the assumption that users would find value in being able to host their own video-streaming site, which at the time could only be easily accomplished by creating a Wordpress site and embedding via Youtube. Just the right amount of minimalism was applied above to create an MVP that allowed Fliggo to not only test if early adopters would sign up, but also if they were willing to pay.

If demand for the product vision was verified (which it was in this case), the next iteration of the MVP could test pricing by increasing or decreasing the monthly fee on the landing page. Such a tweak would be minimalistic by not disrupting the simple interaction of filling out the form while still focusing the learning even further.

While it's cheap and simple, a landing page is not a one-size-fits-all MVP. **Startup**

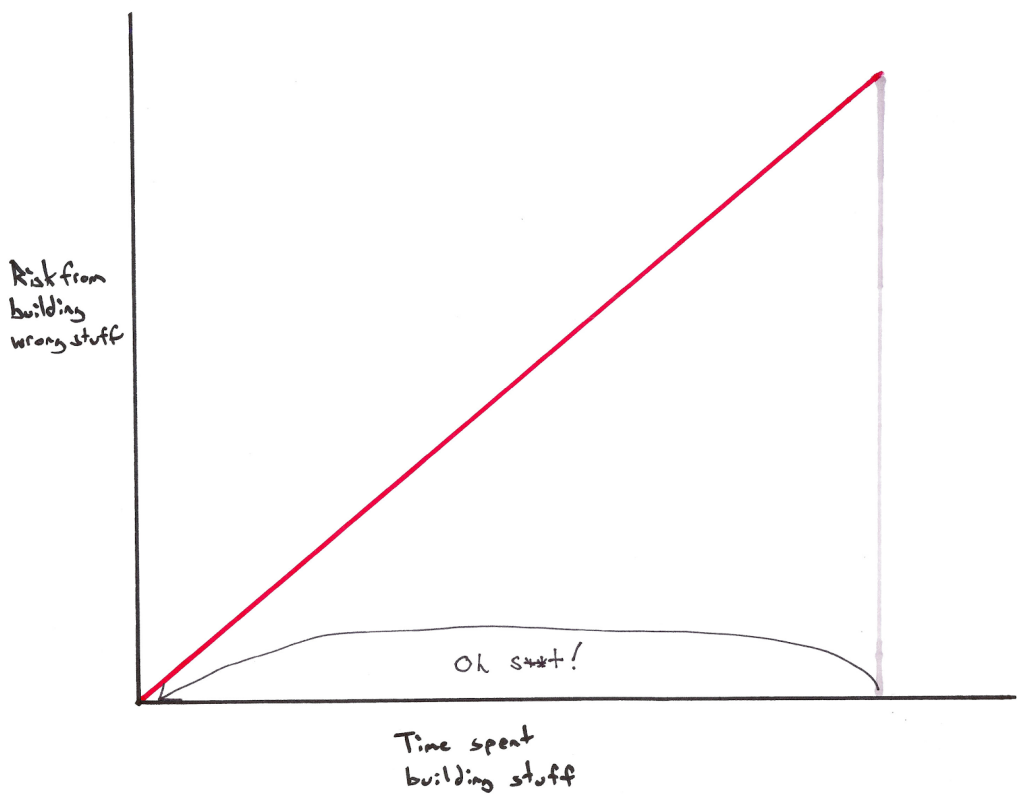
advisor and founder of Extreme Lean TV, Ramli John, warns against taking a uniform approach to MVPs. Relying solely on landing pages is dangerous because you might not able to gather feedback from early adopters who sign up (What are their problems? What would they like to pay?) and lack of signups might not be due to the product itself (perhaps the copy is poor or design is distracting).

Ramli advises other low-cost, high-feedback MVPs such as email, blogs, and video. In fact, as Ramli points out, popular startup and investor matchmaker AngelList tested its networking value by emailing introductions between startups seeking funding and active investors — validating its business hypothesis while still providing a pleasant early adopter experience. By simplifying down to just its unfair advantage, AngelList created the right MVP.

FOCUS ON THE ASSUMPTIONS TO AVOID SCOPE CREEP

An important aspect of MVP minimalism is the time dedicated to development, as this is as precious a resource as money or manpower. Although projects often experience scope creep, this is arguably one of the most comprehensible aspects of designing and developing MVPs.

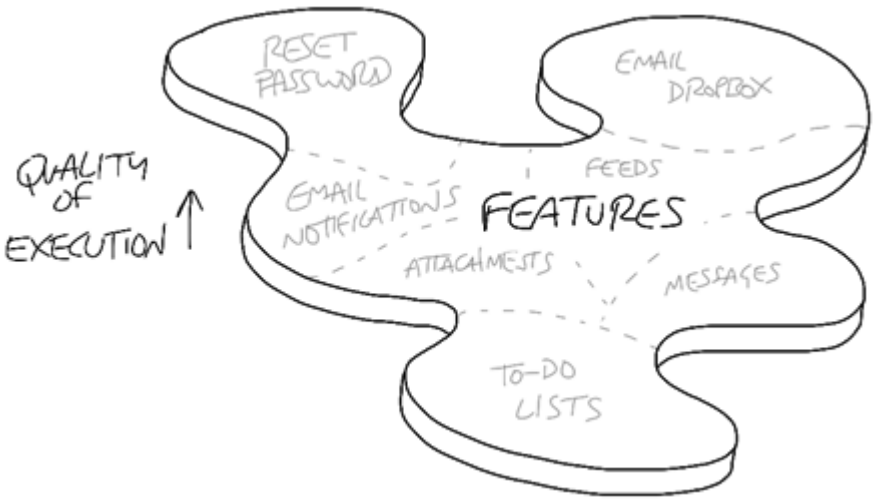
In UXPin, our team collaborates on deciding the amount of time required to complete certain aspects of the product design. Once we finalize the design and compile our comments from UXPin, we can plan out the time to production in more detail in a project management tool like Asana or JIRA.



Source: The Wrong Stuff vs. Stuff With Errors

So how long is too long to spend on developing an MVP? **Rick Screnta, creator of the search engine Blekko,** maintains that it depends on the product. For some web applications like Fliggo, a simple landing page was enough. For offerings with higher expectations and greater coverage and attention, MVP development can and probably should take longer to avoid negative user backlash that arise from an overly simplistic or poorly built product. He cites his company's search engine, which required 3 years for an MVP, as a primary example of this assertion.

A more relevant inquiry may be *what* to build, as opposed to *how long*—as the answer to the former question invariably answers the latter. In the illustration from Signal vs. Noise below, **Ryan Singer, a Product Manager at Basecamp,** suggests that minimalism in MVP scope is achieved by maintaining a base quality of execution and adjusting the number of features accordingly.



Source: What Happens to User Experience in a Minimum Viable Product

KISSMetrics founder, Neil Patel, cites some interesting examples of some notable humble MVP beginnings of varying scope but consistent execution:

- **Dropbox** — started with a 3 minute video for their MVP, resulting in signups increasing from 5,000 people to 75,000 *overnight*—all of this in absence of a real product
- **Foursquare** — started from collecting customer feedback using Google Docs
- **Virgin Air** — began with one plane and one route to validate their assumptions, with more planes and routes added as they refined their business
- **Groupon** — started as a WordPress blog with a widget that sent PDF coupons via email

Virgin's MVP was the most resource-intensive, while Foursquare's MVP was the most lightweight. However, each MVP was only as complex as the assumptions it sought to test. More importantly, you'll notice that each MVP wasn't over simplified to where the value proposition and early adopter experience were sacrificed. Present-day Groupon, for example, still delivers on the value of its MVP by con-

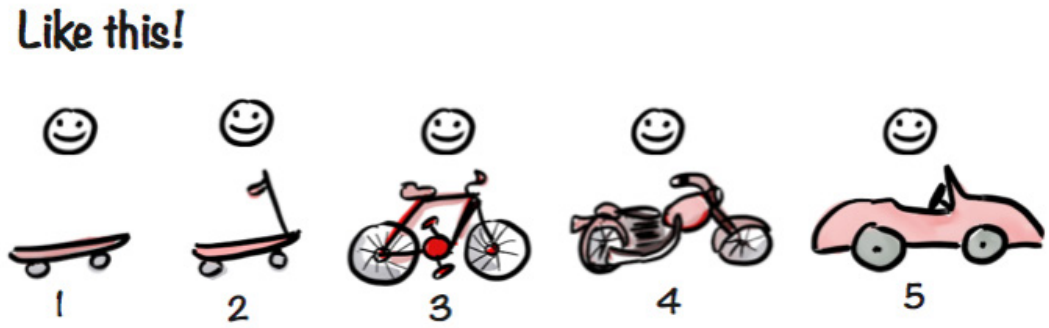
solidating coupons for customers via email — just in a sleeker and more refined interface.

Marcin Treder, **CEO and Co-Founder of UXPin**, believes the best way to determine the correct scope of an MVP is to first understand what an MVP is not. "Unfortunately, the most popular misconception is that an MVP is a minimal product", Treder says. "An MVP isn't the quickest or the most perfect product. Rather, it is a product with minimum development effort that creates maximum value." In order to achieve this in practice, you must create the smartest test you can design to either validate or invalidate the hypotheses behind your product. Learning both what works and what doesn't are important — you can still find focus by deduction.

STAY TRUE TO YOUR PRODUCT DNA

Regardless of the tactic, make sure your MVP strategy stays focused on testing assumptions rather than stripping down features for expediency. As long as your MVP remains true to your unique value proposition, you can always iterate the nice-to-haves based on early adopter feedback.

At UXPin, we made sure we conveyed our core value of comprehensive and collaborative design in all our iterations. Our product philosophy is to use an MVP approach to developing each new feature so that our sprint cycles are as lean as possible.



Source: Stop Overthinking... Just Stop

In the illustration above, the "product DNA" of the car is its efficiency at getting the user from Point A to B versus just walking. Each iteration expands the scope of the product with more features, but the unique value of faster transportation remains consistent.

The MVP in this product development example could actually be as simple as an online poll asking people if they want a quicker way of getting around town — it is the easiest way to test the unfair advantage with minimum scope (cost, design, and engineering). Based on early adopter feedback, you could then design a skateboard and iterate until it becomes a car.

Gagan Biyani, co-founder of Udemy and Sprig, believes that your first MVP needs to test your core value while further iterations should test new hypotheses. "MVPs should be focused on being a minimum viable test for hypothesis X, not just a product," says Gagan. "Successive MVPs that test different theses will let you launch faster and better".

> *"MVPs should be focused on being a minimum viable test for hypothesis X, not just a product"*

Share Quote

For example, one of Udemy's early MVPs was a $20 online course filmed by the co-founders which was used to test the hypothesis that customers would pay for a high-quality video course. At Sprig, the MVP involved the founding team running a one-night meal service to validate the hypothesis that food could be delivered in under 20 minutes. Just like the skateboard illustration, the first MVP validated the core assumption behind the business, which allowed future iterations to test more complex hypotheses.

KEEP IT SIMPLE AND EXPERIMENTAL

MVPs succeed by testing if there's market demand for an alternative without obsessing over features. By limiting your scope to testing just the core value of your product, you give yourself room to fail without breaking the bank. Remember, the goal of an MVP isn't getting it right, it's maximizing learning with minimal effort so you don't go down the wrong path.

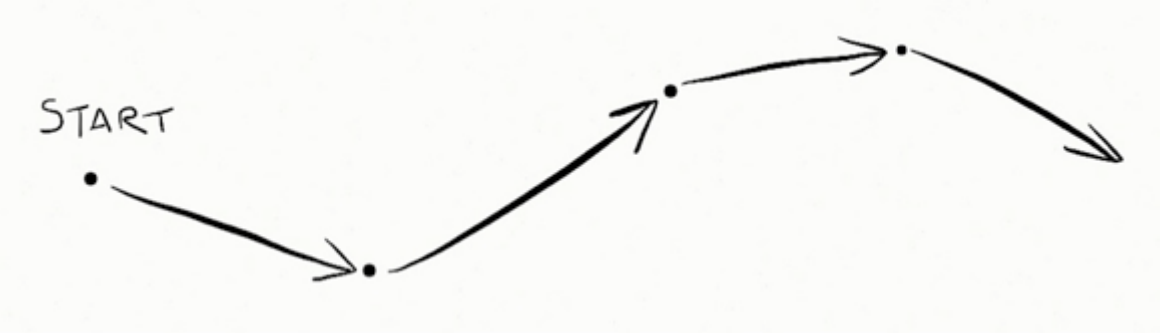
CHAPTER THREE

THE V IN MVP: EXPERT TAKES ON PRODUCT VIABILITY

How to think about viability before building your next product

When it comes to MVPs, it's easy to get tunnel vision and zoom in on just the "minimal" part.

But a product that works isn't enough. It needs to be positioned for long-term success. Otherwise, you might as well launch your MVP into outer space.



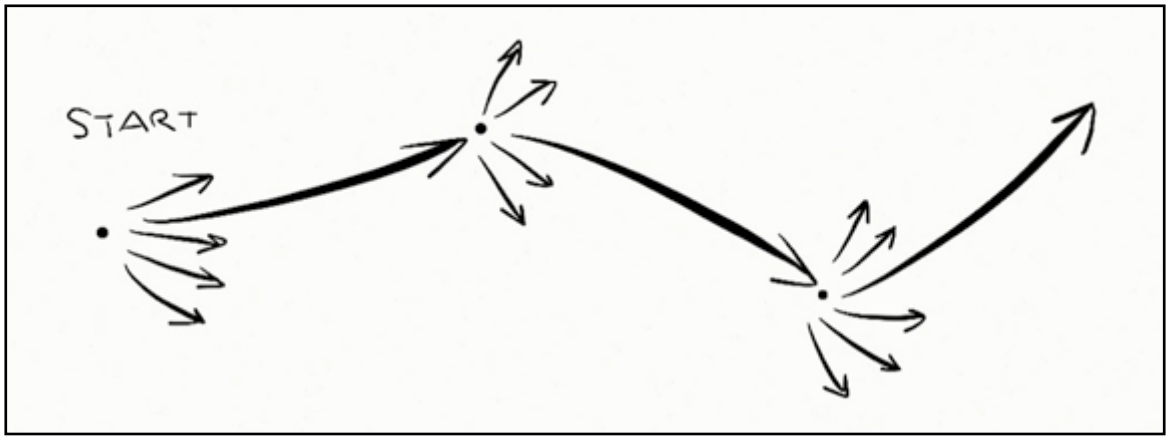
Source: Don't Let The Minimum Win Over Viable, HBR Blog

As shown above, the prospect of pivoting down the line can lead companies to iterate without questioning their hypotheses. At every point in the process, teams focus on adding, removing, changing or tweaking features rather than the purpose for doing so. This leaves them with a polished product at each stage that may not be viable because it was built for the wrong business reasons. Hence, the need to keep pivoting.

In this piece, I will help you avoid that mistake by explaining how to explore all options when building your MVP, the difference between product and business viability, and how to engage your team to ensure viability.

ESTABLISHING AND SUSTAINING VIABILITY

In UXPin, I can create multiple product prototypes quickly with the feature sets I think customers want. Then I can personally show them, test each variation and get their feedback before taking my MVPs and iterations to production. This keeps our team centered on a well-designed business solution instead of just a technical marvel.



Source: Don't Let The Minimum Win Over Viable, HBR Blog

Unlike the first illustration, the diagram above shows a flexible approach to MVPs whereby product teams focus on breadth rather than depth at each iteration to make sure they're choosing the optimal product path. Once the direction is chosen, then the team dives deep to execute on the product direction.

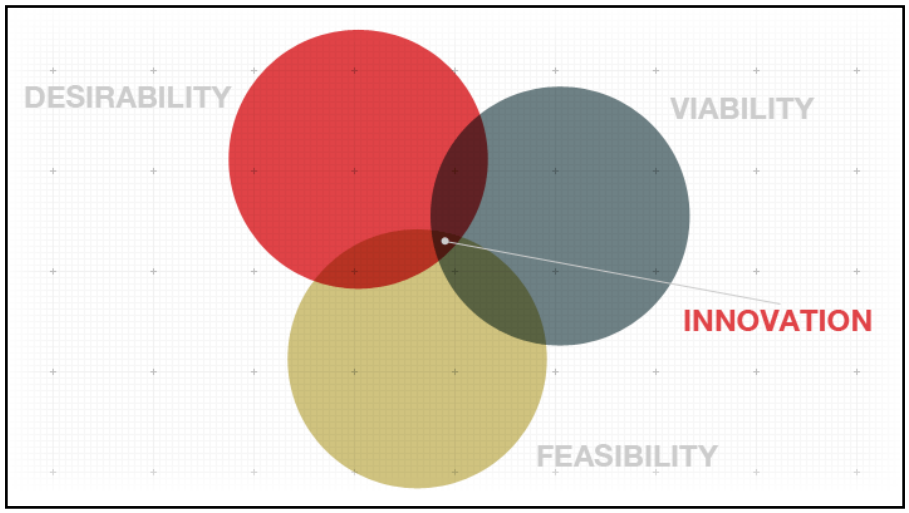
Even with an Agile methodology, concepts can gain disproportionate traction because of the team's past investments and labor. However, ideating on different user experiences and needs lets you break free from previous commitments to see the whole landscape. Focusing the team on batches of small iterations lets you more frequently check if you can build the product, whether customers want the product, and if they're actually willing to pay for it.

According to Steve Blank and Bob Dorf, **both notable entrepreneurs**, a web service can use multiple landing pages to test viability of different solutions. As mentioned, an online payment service can be developed into 3 prototypes: FastPay, EZPay, and FlexiPay. FastPay addresses issues around speed, EZPay addresses ease of use, and FlexiPay addresses flexibility. Instead of releasing an MVP that dives deeply into just one customer problem, they minimize risk by diversifying laterally and then testing to find the best candidate.

Diversity and flexibility helps teams see beyond just functionality into the real customer problems. The above model prevents you from modifying the product when tweaks to your assumptions about the market are needed.

UNDERSTANDING PRODUCT VS. BUSINESS VIABILITY

Viability is one of the most contentious and misunderstood aspects of MVPs. It's the primary reason businesses and business units fail. And it's arguably why there's such a frenzy among investors, venture capitalists, entrepreneurs, press, social media elite, and anyone else in tech when early-stage products show great viability — it just doesn't happen as often as we'd hope.



Source: Innovation 101

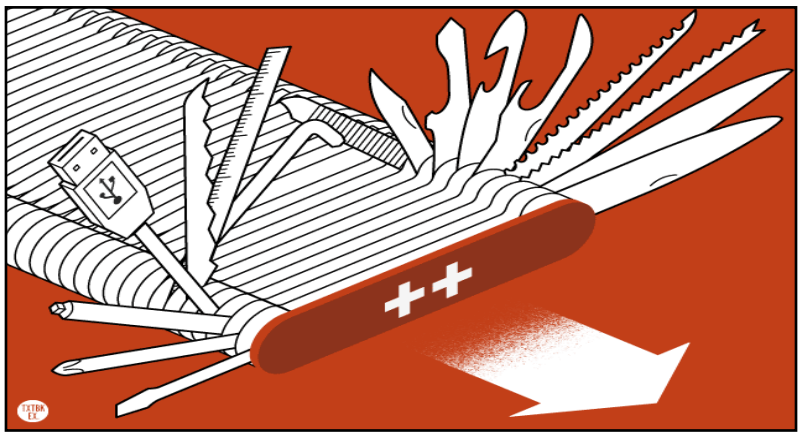
According to **Christina Wodtke**, **former General Manager at Zynga**, we need to look at viability through a business lens to ensure our MVP stays focused on the market. "You need to Baby Bear it so it's just right. And typically that's the smallest of all efforts." For many companies, it can be tempting to build the perfect viable product simply because they have the resources to do so. Unfortunately, this only leads to disappointment down the line as they soon find out that

market conditions won't allow for scalability beyond just the MVP.

To help you ideate and iterate smarter as you develop your MVP, let's compare the definition of a viable product versus a viable business.

DETERMINING PRODUCT VIABILITY

Product viability is defined by feasibility, which requires careful consideration. Political, legal and other market or product-specific factors must be examined along with technological factors. To name a few, tax policies, health & safety laws, and limits of existing technology can all limit early products. For example, Airbnb's feasibility is limited by zoning laws dictating how long paying guests can stay and Spotify had to remove a portion of tracks due to licensing issues only a year after its founding. While both companies are doing well now, you can bet those political and legal issues posed setbacks. The wonder-knife below perhaps exemplifies feasibility: if the right conditions permit someone in the world to build it, then it's technically feasible.



Source: Textbook Example

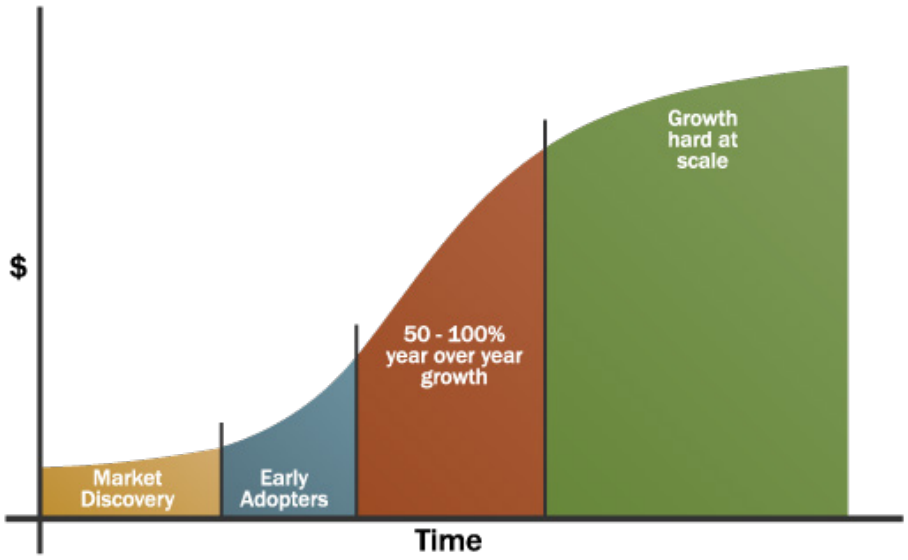
Now let's apply this thinking to a hypothetical situation. A company wants to see if unmanned drones would be valuable for collecting data on crop health (i.e.

whether plants are diseased, if there's an insect infestation, etc). Amidst a hundred-page spec document, the team verifies that the right spectrometers exist to detect plant chemicals, the data can be imported into Excel, and FAA regulations aren't overly restrictive. The project is deemed technically viable — just like ethanol-powered cars and solar power.

But the MVP could ultimately fail because it does not test the viability of business by verifying market and company capabilities. Can the team cover the cost of long-term maintenance for the drones? Is the profit margin wide enough to justify the R&D? The key to testing viability is keeping sustainability in mind, not just if a product can be built using any combination of technology and under the right legal, political or other circumstances.

DETERMINING BUSINESS VIABILITY

Going from early adoption to sustainability is one of the hardest stages businesses experience. Unlike product viability, there are multiple considerations when thinking about business viability. I've expanded on these points below.



Source: [The Brown Swan Theory]

1. DESIRED BY THE MARKET

As we discussed above, perceived market need is often followed by single-minded development without measuring and refining the vision. If we continue the analogy, the UAV team failed to answer perhaps the most crucial question of all, which is whether farmers even care about the data in the first place.

Joel Gascoigne, Founder and CEO of Buffer, suggests that the MVP's sole purpose is to validate learning about what the market currently needs. In our example above, the team could have tested the market by renting an airplane, taking photos and processing the information themselves, and then getting feedback from farmers. In some cases, you might not even need to build a physical MVP. Landing pages can be the most cost efficient means of testing demand — certainly, if a sizeable amount of potential users sign up in anticipation of the product, then the product is at least potentially viable.

David Aycan, Design Director at the esteemed design and consultancy firm IDEO, expands on this distinction between building products that are essential for customers versus products that are technically feasible. The "Minimum" in MVP connotes a correct set of features that are important to the customer, and is not related to ease of technical execution. "Viability" is thus a measure of the product's ability to focus on addressing the customer's core needs in a revolutionary way. "Don't let the minimum win over the viable", as Aycan aptly advises.

In "Top 10 Ways to Test MVPs", we'll actually go into more detail on conventional and unconventional tactics you can use to test desirability.

2. BUILDABLE BY THE COMPANY

Stella Fayman, a noted expert on entrepreneurship and lean startups,

defines viable as getting the job done — leave out all the bells and whistles. Buildability is achieved by seeing technical feasibility through a minimalist lens, as the main purpose of the product at this stage is to test assumptions regarding your product. To this end, asking yourself the right questions are instrumental to being smart with resources; therefore, optimizing buildability. Here are a few:

- What core functions does our MVP need?
- How do we know if our MVP is successful?
- What do we hope to learn from performance of the MVP?

According to **Neil Patel**, **co-founder of KISSMetrics** and **advocate of lean product development**, working efficiently in small batches is the key to building out an MVP. By working in small sprints, team members become more efficient at evaluating what features are core to the vision and what are extraneous. Frequent design reviews and code checks keep resources focused on what will get the job done and makes your team better at what they do. This goes a long way towards catching red flags — for example, creating code that depends on a certain configuration and then having that configuration change. Working in small batches reduces development cost since the above mistake could cripple the ROI on your MVP (the number of signups you get may be irrelevant if the cost of reworking the code is astronomical).

> *"Fast and flexible is more important than slower and specialized when building your MVP team."*

Share Quote

Patrick Neeman, founder of UsabilityCounts.com, advises that you select multi-talented team members to work in small batches since it takes a village to build a product. Your MVP team should ideally consist of people who can tackle product management, interaction & visual design, development, content creation, and QA. While you don't need one person for each of these roles, the

responsibilities should all be covered. “Fast and flexible is more important than slower and specialized when building your MVP team” says Patrick. “If your lean MVP team can’t wear multiple hats, then that’s a huge liability.”

3. PROFITABLE FOR THE COMPANY

“Asking customers to buy something will yield more relevant insights than simply asking them if they will buy something”. While surveys can help check if customers will buy, placing a hurdle in front of the customer will help assign a dollar value to your idea. For instance, you can place a payment button for a set amount on your landing page. Once they click the payment button, notify them the product is still in alpha stage but they will be able to beta test. The people who completed the process are now ideal early adopters since they’ve shown they have enough pain to make them open their wallets.

Marcin Treder, CEO and cofounder of UXPin, believes that MVP profitability boils down to quality of feedback. “8,000 people who give you their email address isn’t worth nearly as much as 30 people who are willing to pay now and offer input.” says Treder. “The goal of an MVP isn’t quantity but quality. What you learn from those 30 people can help you monetize everyone else.” Getting early adopters to cross this “penny gap” to become paying customers is perhaps the hardest part since you’re not just competing with what’s similar, but also with what’s free.

> *“8000 email addresses isn’t worth nearly as much as 30 people who will pay now and offer input”*

Share Quote

Neil Patel, cofounder of KISSMetrics, detailed a few tips to help cross this pay bridge which we’ve adapted for our own purposes below:

- ***Inform customers that you’ll need to start charging*** — Do not hide or blind-

side them with a fee when earlier iterations were free. This can drive away your early adopters.

- ***Be honest about why you're charging*** — Most people understand will understand that an MVP, even in a free market, won't be free forever.
- ***Be transparent about costs and profit*** — Honesty is a great way to build trust and revealing profit and salary helps people see what's needed to sustain the vision.
- ***Start lower than your target fee*** — Explaining to early adopters that you're charging below the profitable rate is a nice way to ease them into paying for your product and primes them for future price increases.

Alternatively, you can also "sell first, build later" by using crowdfunding sites like Kickstarter.com. Unlike landing pages and other MVPs, this tactic can gauge up-front what customers are already paying for your idea — information that's critical since setting the wrong price after you launch could cripple your profitability. Not only will a successful crowdfunding campaign earn you a community of early adopters, it also sets early expectations around pricing since different levels of donations receive different levels of product.

> *"Do whatever it takes to get something into someone's hands. You'll learn more from in-the-flesh customers than any degree of theorycrafting."*

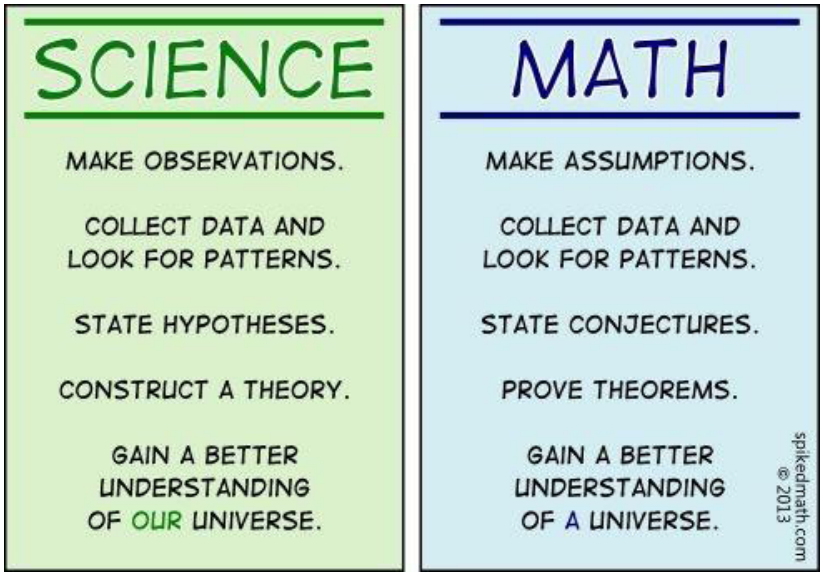
Share Quote

John Saddington, Partner at startup accelerator The Iron Yard, verified that people were willing to pay for an app that gave them more creative control of filtered photos than Instagram or Facebook. By using Kickstarter, John actually raised 113% of his $50,000 goal for his app Pressgram. It worked so well that

he plans on taking the same approach to validating its desktop iteration. “Do whatever it takes to get something into someone’s hands. You’ll learn more from in-the-flesh customers than any degree of theorycrafting,” advises John. “That’s what an MVP is all about.”

HAVE A METHOD FOR THE QUALITATIVE MADNESS

When it comes to MVP success, it’s important to analyze qualitative and quantitative data. Quantitative data is often more straightforward while qualitative data can be less actionable. We’ll show how to better understand qualitative data to inform your quantitative decisions — the two are undoubtedly intertwined.



Source: IFLScience

Lean Startup coach Tristan Kromer believes that evaluating the data of your MVP is just as important as building the product part. In fact, there’s actually four parts to an MVP. “The four parts to an MVP are Customer, Channel, Value, and

Relationship. If you're missing one, your feedback loop is broken," says Tristan. "You've built your product. But did you build in metrics or a marketing channel? Building doesn't always mean building product."

For more actionable insights, Shopify and Mashable have succinctly outlined ways you can evaluate the viability of your product ideas and then make your product viable. While Shopify's guide is tailored for the physical products which its merchants sell on the e-commerce platform, many of the same viability criteria apply to digital products.

Assuming you've already got a product out the door or are working on one, here are 4 steps you should take:

- ***List 30-40 critical success factors*** — related to your product, customers, transactions, your product category, the environment and more
- ***Gather and report customer input on these factors*** — and have each team member rank them from 1 to 10 to determine how viable a product is at a particular time
- ***Combine the scores to get a team score for each factor*** — then discuss the rankings in detail face-to-face to better qualify each score, and get the entire team on the same page
- ***Plot the scores over time*** — then you'll have a better sense of how your success factors (i.e the viability of your product in certain respects) changes over time to gauge your team's performance in reaching its goals

The answers from the questions above will go a long way towards helping everyone from engineering to design understand that viability is an ongoing goal rather than just a process.

GIVE YOUR PRODUCT A LIFE OF ITS OWN

To think about viability in the simplest terms, let's consider that the origin of the word "viable" actually comes from the Latin word for life.

Unfortunately, the common thinking behind MVPs usually leans towards "What is the least we can do to create something that won't fail?" Instead, ask yourself "What can we deliver that will grow into a life of its own?" With that in mind, go forward and build your MVP knowing that it is a seed which must grow into a profitable product.

Source: Homegrown Tips

CHAPTER FOUR

P IN MVP: EXPERT TAKES ON PRODUCT QUALITY

How to think about quality when building your next product

What is the right level of quality to include in an MVP? It's not just about how much time you spend on building the product or how much lipstick you put on a hoggish product. For you to get this right, you have to really understand your users, how they'll use your product, and what will make them really happy — and really pissed off.

Before going too in-depth on product quality, let's first look at the overarching dimensions of what make quality products.

Source: Moz, "7 unlikely recommendations for startups & entrepreneurs"

8 DIMENSIONS OF PRODUCT QUALITY

Before rallying your team around quality and delivering quality MVPs, it's important to first think about the dimensions that comprise a quality product.

David A. Garvin, a Harvard Business School professor, wrote extensively about approaches to defining product quality and the overarching dimensions of product quality. Although this was written two decades ago, much of it still applies to contemporary high-tech web and mobile products. Through his analysis of transcendent, product-based, user-based, manufacturing-based, and val-

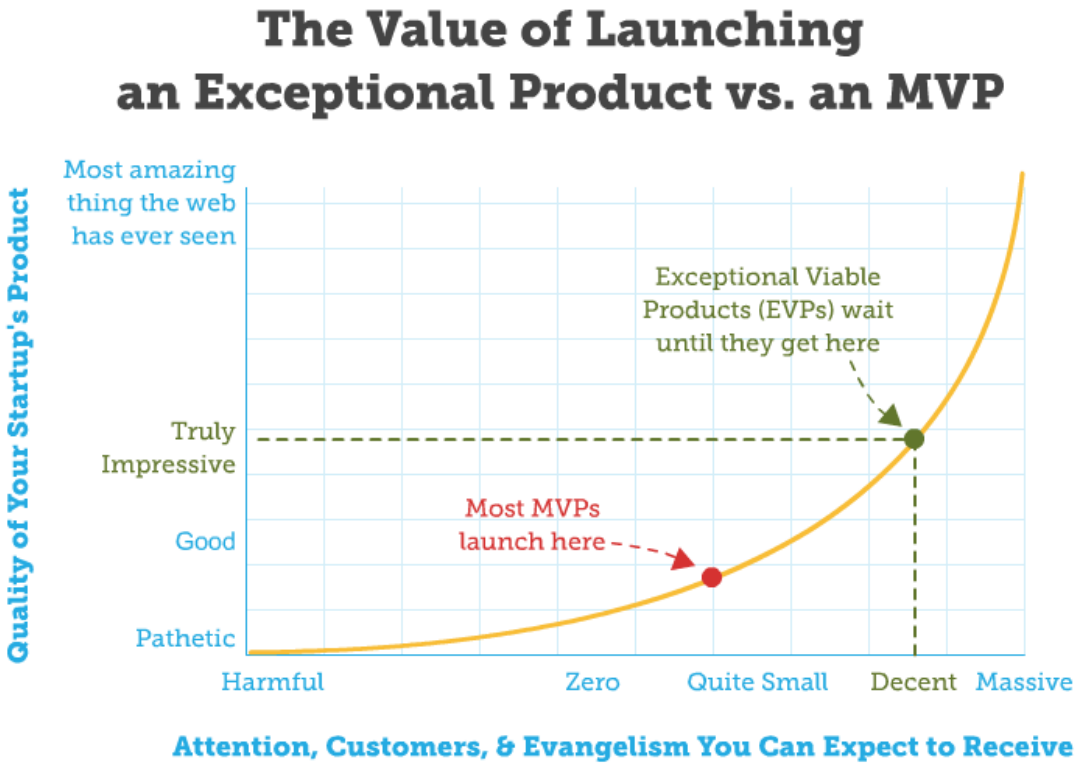
ue-based approaches to defining product quality, he enumerated 8 dimensions of product quality I've summarized below:

- **Performance** — a product's primary operating characteristics
- **Features** — the "bells and whistles" that supplement basic functioning
- **Reliability** — the probability of a product malfunctioning within a specified time period (reliability may perhaps better be defined as the probability of a product not malfunctioning)
- **Conformance** — the degree to which a product's design and operating characteristics meet established standards
- **Durability** — a measure of product life which also has an economic component
- **Serviceability** — the speed, courtesy, competence, and ease of repair of the product
- **Aesthetics** — how a product looks, feels, tastes, sounds or smells
- **Perceived Quality** — the image or reputation of a product

He also elaborates on how product quality may impact product profitability, advertising, market share, price, and costs — all important factors that impact business and product viability. Unfortunately, it only scratches the surface but I'll definitely share more insights as I come across them.

PLAY TO EMOTIONS — DELIGHT EARLY ADOPTERS

Rand Fishkin, co-founder of Moz, has a different take. In general, he thinks MVPs suck and companies should aim more for what he calls an **EVP — an Exceptional, Viable Product.**



Source: Moz, "7 unlikely recommendations for startups & entrepreneurs"

While he's a long-time proponent of building something small and minimal then iterating on it, he's seen a lot of MVPs launch that rarely produce exceptional value and are, therefore, rarely praise-worthy. Many experts like **Steve Blank, a serial-entrepreneur and author / lecturer on MVPs,** insist that influencers and amplifiers — not just any user or customer — are critical for getting traction, and if you can't impress them, then it will be far more difficult to get proper traction — and, like re-skinning a pig, there's only so many times you can re-launch a product. This is becoming true even at later stage companies with massive

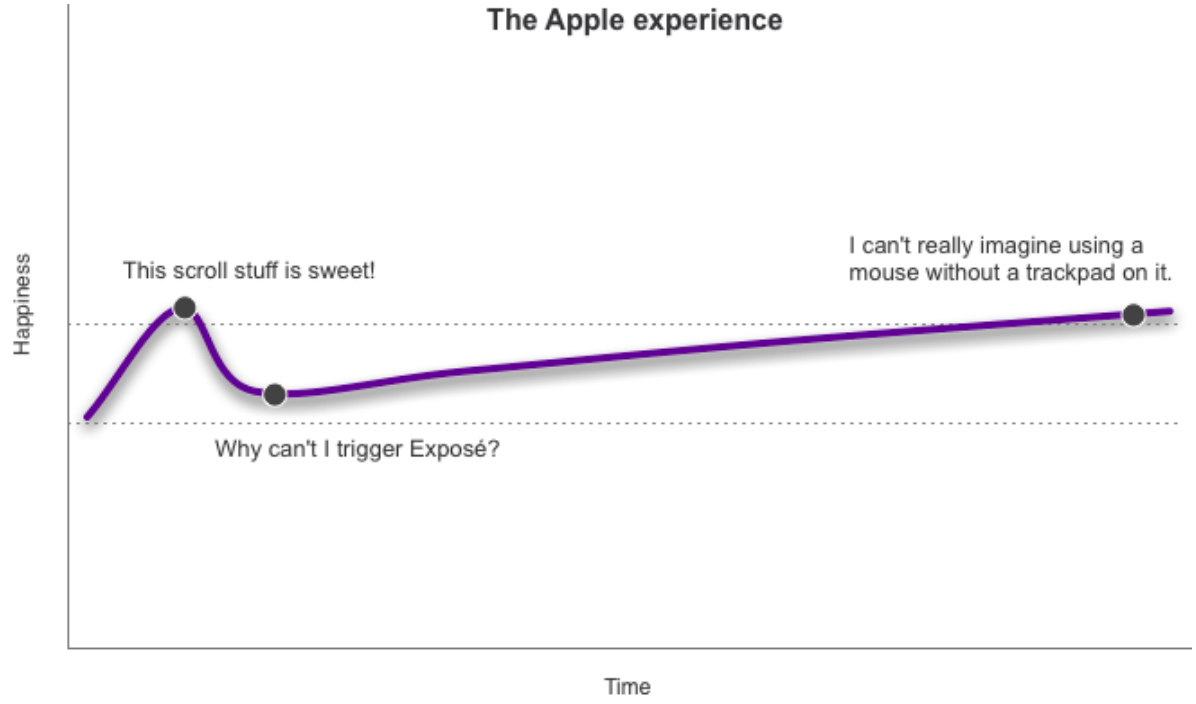
existing distribution channels, and with serial entrepreneurs and executives with sizeable followership. Although they may help with initial visitors and signups, users and customers — especially influencers and early adopters — are hardly loyal to mediocre products. Products that fail to hit the "A Ha" moment will churn their user and customer base as quickly as they acquired them.

In practice, Rand suggests making your MVPs in-house and dogfooding them internally and with a few customers. Gather feedback and iterate until the first internal and external users find that "A Ha" moment, then release it to the wild as an EVP. This may take an extra 30-90+ days to reach this point but, in his opinion, it's well worth the wait. "The MVP has biased too many startups to think about what's minimum rather than what's viable," advises Rand. "Don't be minimum, be exceptional."

> *"The MVP has biased too many startups to think about what's minimum rather than what's viable."*

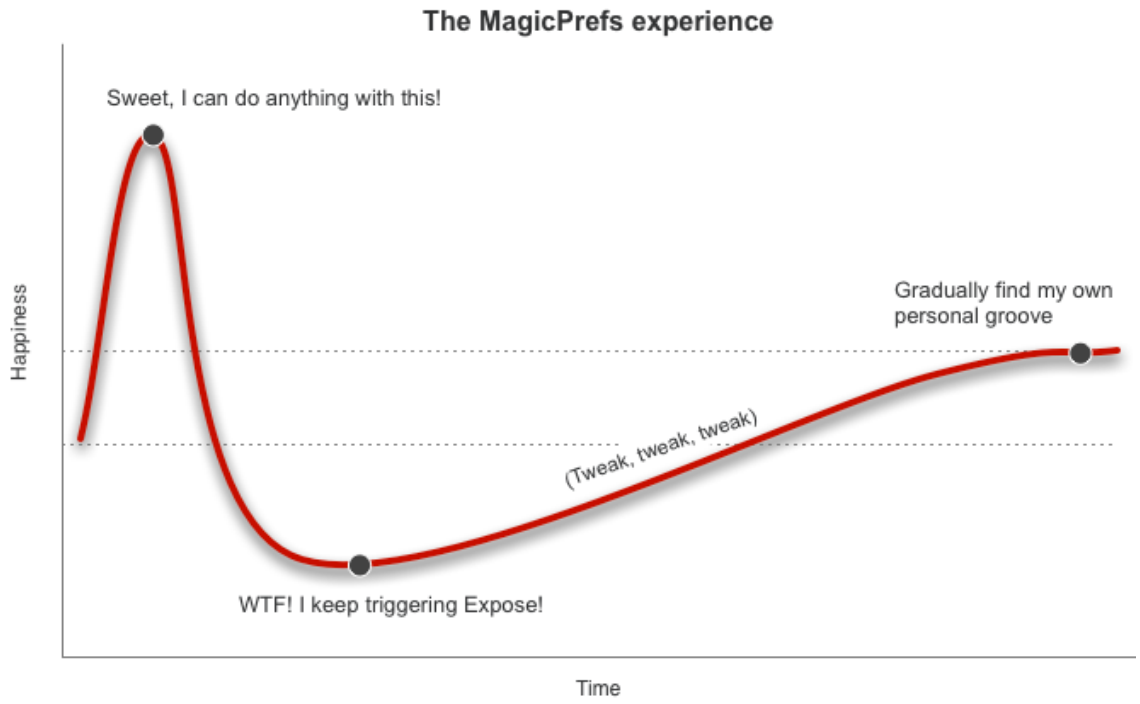
Share Quote

To illustrate my point, let's look at a user's experience of Apple's Magic Mouse compared to an alternative product, MagicPrefs. In the diagram below, Brian Donohue, describes his experience using Apple's mouse. While his initial expectations of the Magic Mouse aren't higher, Brian ends up loving it after experiencing the value of a feature he didn't even know he needed. In his eyes, Apple creates a truly delightful technology — a mouse with a trackpad that can support a bunch of different gestures and actions — by limiting the functionality to a small percentage of what it's capable of (i.e. displaying dramatic design restraint).



Source: [Apple's Magic Mouse: An Example of Taking "Design Restraint" Too Far?]

Meanwhile, MagicPref sets higher expectations for what users can do with their product and, therefore, runs the risk of severely disappointing users when they do actually remember how to use all of the features. Eventually, committed power users will adjust their habits to get into a personal groove. But odds are that t everyday consumers will just opt for a less complex product like Apple's Magic Mouse that has a delightful twist compared to other alternatives.



Source: *Apple's Magic Mouse: An Example of Taking "Design Restraint" Too Far?*

At UXPin, every team member can review and revise our product designs throughout the process in real-time from anywhere. Having the right eyeballs looking at every detail of what we're trying to accomplish with our MVPs, my team is able to deliver higher-quality products that our users want. Customer support, sales, marketing, design, engineering, and product teams and, of course, our CEO are all able to give their input without any hurdles. Of course, we still use whiteboards, sketches, presentations, and other tools depending on how we need to communicate with one another. But it all gets recorded back in UXPin. It's pretty fantastic.

ADD LOGICAL VALUE — HELP THEM DO SOMETHING

Josh Puckett, product designer at Dropbox, states that assessing quality level is contingent on the goals being set for the MVP. "Design has a single goal for

an MVP," says Josh. "Help users easily understand the value your product adds to their lives. Don't distract from this. That is to say, if the quality level (or lack thereof) makes it hard for the user to accomplish the goals set forth, then not enough quality is present in the MVP offering. From his perspective, it's often helpful to put your MVP through a quick test to see if more time needs to be spent on design:

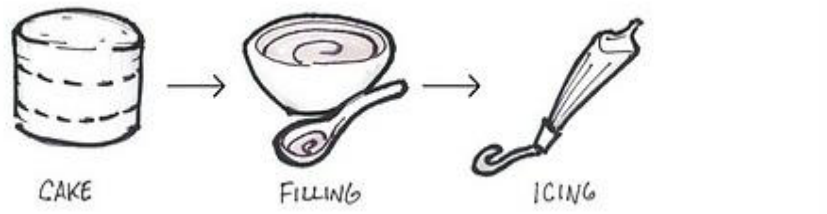
1. Is there friction due to poor design that impedes users from accomplishing your goals?
2. Is the design and user experience at level that you would feel comfortable shipping to all of your users, as a final product?

If your MVP fails either litmus test, he suggests you slow down and revisit the product design. He's not saying the product has to be pixel and animation-perfect. Rather, he's trying to avoid obvious pain points and poor interactions that may corrupt the insights you're supposed to gain from an MVP — if users are annoyed or confused by your product, their actions in the product won't make much sense... if they even stay that long.

Brandon Schauer, CEO of Adaptive Path, shares a framework he's used over the years to help teams think through a successful customer experience in the short-term and long-term — The Cupcake Model. "The MVP is one of the most referenced and least understood concepts in modern product development" says Brandon. "Embrace the reality, not the hype." He details two drastically different approaches, the Dry Cake and the Cupcake model, highlighting the superiority of the latter approach.

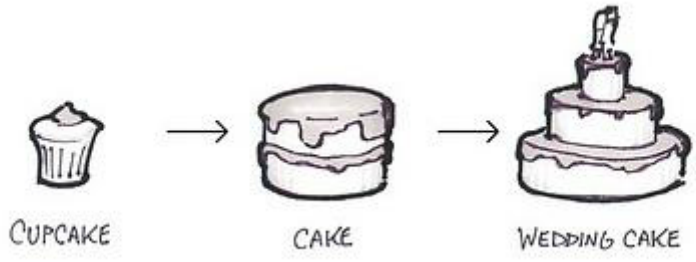
In the Dry Cake model, product teams start with a very basic product that may not be very interesting — like a plain dry cake. Then they add new features such as icing or filling to get a more interesting and complete end product. This is the approach many organizations take, unfortunately. While it makes great sense

operationally, it's problematic from a competitive and customer perspective. As he puts it, "cake with no filling or icing isn't that appealing. Plus, anyone can just make a cake."



.*Source: the cake model of product planning*

In the Cupcake model, product teams start with a smaller yet complete product that is likely more desirable. It has cake, icing, and filling — a tasty treat for anyone looking for a sweet treat that will get the job done between meals. Think about the craziness of gourmet cupcakes: three tiny cupcakes cost just as much as a large, plain cake in the store even though they're half the volume or less. People want a complete product, and they'll pay for it. It also sets you apart from the bland or chaotic alternatives.



Source: the cake model of product planning

Stephen P. Anderson, UX consultant and author of "Seductive Interactive Design", believes the cupcake model of creating a complete experience is crucial since many companies compete in mature markets. "The MVP has been (ab)used to justify sub-par product experiences where quality is sacrificed for speed" says Stephen. "If you're building something entirely new, a buggy prototype can be good for learning. More often than not though, you'll be competing in mature spaces where others have already set the bar for quality and a poor experience

can produce misleading data."

Knowing the market and being honest about your MVP is the best gut check on quality. If you're truly first-to-market, then you may be able to release something a little clunkier if the idea is innovative. On the other hand, if you're coming in as the underdog, don't even think about skimming on quality.

LEVERAGE NETWORKS — DON'T RELY ON THEM

The importance of this aspect of MVPs can not be stressed enough.

In fact, **Gerard J. Tellis, a professor at the USC Business School,** believes that product quality has become so important in recent years, that not only has it shown to be more significant than network effects but that network effects actually reinforced quality by driving users and customers to quality, even superior, products. Generally speaking, he noted that brand loyalty — and, therefore, network effects — were so weak that market dominance tended to shift every few years in various industries.

Marcin Treder, Co-Founder CEO of UXPin, agrees with this sentiment but sees more products requiring network effects to exist in the first place. "The caveat is that networks are becoming increasingly important to many contemporary products such as ad platforms, social networks, peer-to-peer marketplaces, transportation, payments, and so forth," Marcin states, "Nevertheless, if a network is core to the product, then it would be attributed to the product quality as opposed to a distinctly different phenomenon."

Fundamentally, you need to capture the heart and mind of a single user before you can make their friends, family, peers and broader network care about you

themselves. Beyond rigid buying processes, security, and user management constraints, many business applications traditionally haven't been that viral, in part, because they failed to make the end user care — the products were far from pretty or easy to use, and everyone could forget about them by 5pm; no big deal. That's now changing but mostly because business apps are building for the user as much as for the overall business. Remember that every time you have a meeting about increasing your K-Factor.

QUALITY VARIES BY MVP PURPOSE

There are many technical and design considerations when building a product. There's no one-size-fits-all design and engineering formula for MVPs.

> *"Bad code only becomes technical debt when people use your MVP. If you pivot and build a different product, then it never becomes debt."*
>
> 
>
> Share Quote

Kevin Dewalt, Founder of SoHelpful, makes an interesting point regarding an MVP's quality as it relates to technical debt. He asserts that technical debt (e.g. writing poor code or designing poor functionality) can be acceptable since code, features, or design may be temporary for most MVPs.

For instance, if you're building an MVP to validate assumptions with a clear intent on discarding the offering later to build a more robust offering, then suboptimal code and/or design can be an advantage in getting to market faster. "Bad code only becomes technical debt when people use your MVP," says Kevin. "If you decide to pivot and build a different product, then it never becomes debt."

FOCUS ON A COMPLETE PRODUCT INSTEAD OF COMPLETE FEATURES

As you can see, product quality is far more than aesthetics or technical breakthroughs.

It has far more to do with deeply understanding your user, being considerate of every aspect of the product you want to build, and being able to capture the loyalty of individual users (as opposed to vast networks of followers) through adding real-life value and delighting the user. This requires building a complete product in version 1 that can be expanded and improved continuously as opposed to building a partial product and adding distinct layers on top of it.

Your ability to think roughly a few versions ahead will make life dramatically easier — but remember, you'll always need to adjust future versions as you learn from the prior MVPs.

CHAPTER FIVE

BALANCING PRODUCT UX AND LEAN EXECUTION

Dealing with these competing priorities at every product stage

What matters more: a killer UX that makes people want to use your product, or shipping quickly the things people want and staking down a huge share of the market?

Source: Viral Heat

If the UX is bad, people won't want to use it. On the other hand, if someone else gets there first, people are happy to use what is available and help to improve it with feedback as it grows. People have been struggling with these opposing interests forever so we thought it important to outline a better way to think about the problem, depending on your company's stage.

It's important to remember that this is only a framework and that your actual course of action will be as personal as your business and products. But if you're simply trying trying to "wedge in" design excellence while rushing your product releases to market, then you're already on the wrong path.

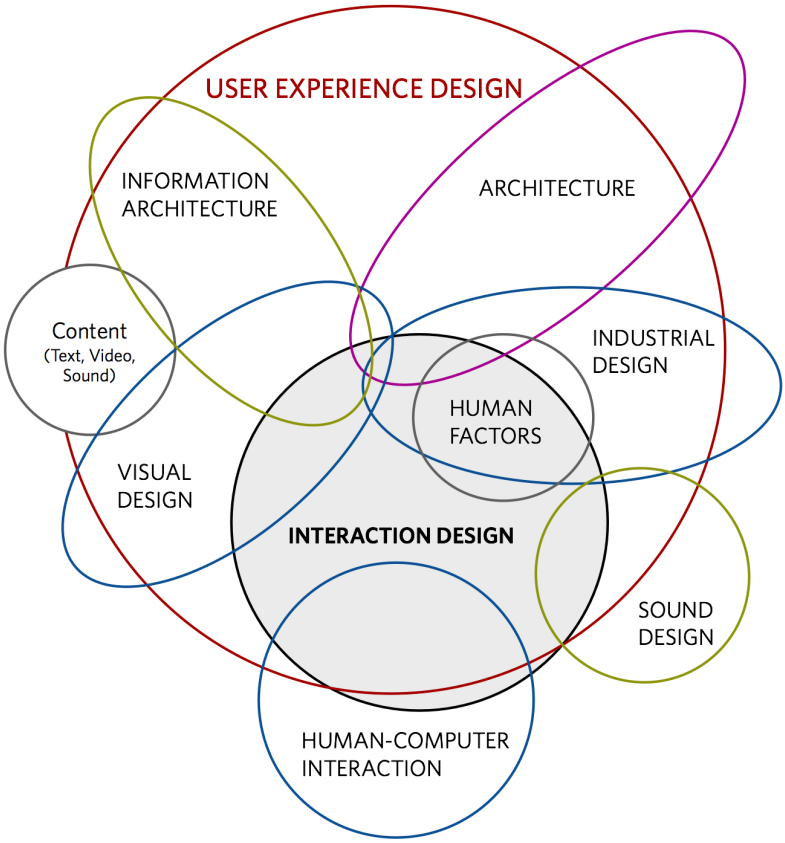
COMPETING PRIORITIES: EXPERIENCE VS. EXECUTION

EXPERIENCE

Great UX is absolutely critical, there's no question about that. **Q Manning at Code Mag** sums up his thoughts on the matter rather well:

"Finding a well-developed app isn't that difficult. Many low-rated apps are responsive and bug free. So what do these top apps all have in common?... The top-selling apps have a fantastic user experience (UX). The best apps do more than accomplish their goals; they transport users into a preternatural state of clairvoyance, where each tap is intuitive and never requires second guessing."

That sounds pretty sweet in theory, but UX rarely reaches that state of perfection. We debate this regularly at UXPin while reviewing UI designs, prioritizing features, planning the product roadmap, and so forth for our wireframing and prototyping app. UX involves how people use your product, what they experience when they do, and how they feel about that. It's such a broad definition that few companies think about it the same way — see one of the many convoluted UX frameworks available below.



Source: Kicker Studio, "The Disciplines of User Experience"

EXECUTION

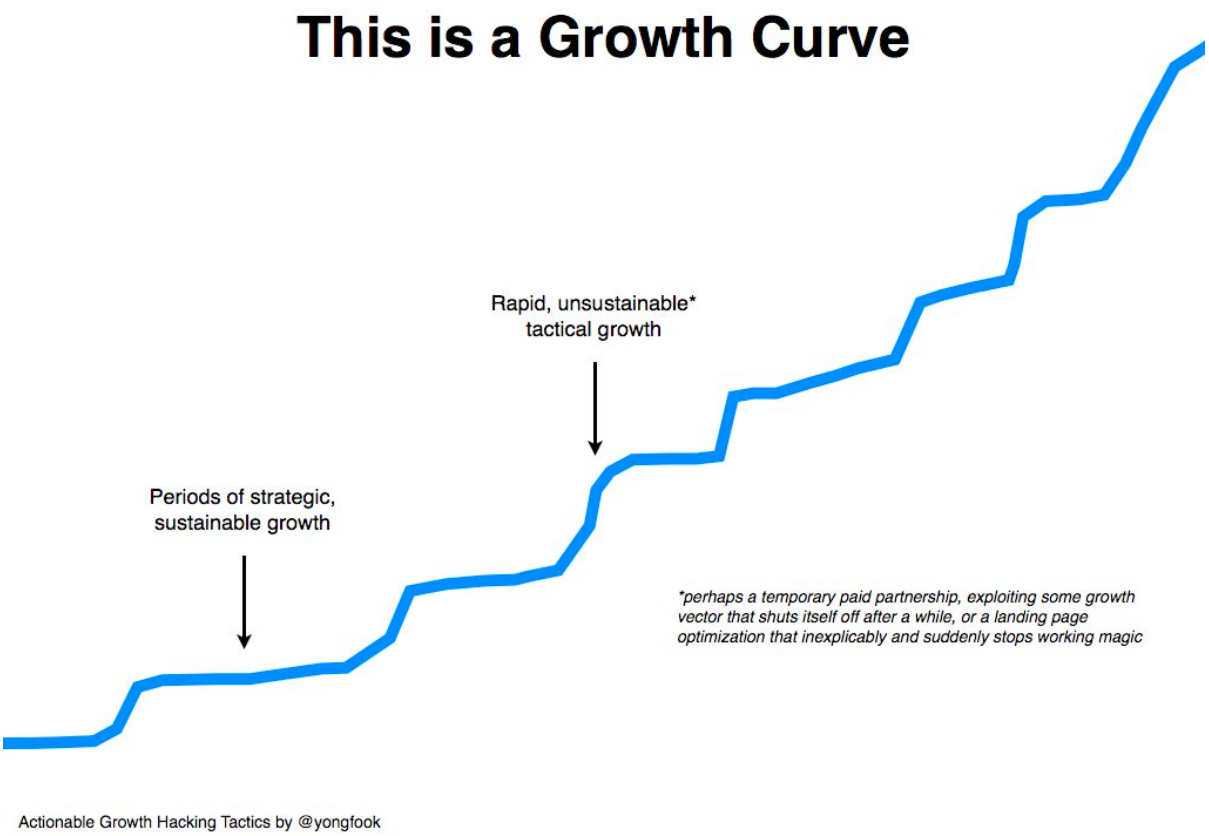
The opposing interest is called, in the language of the Lean Startup model, the Minimal Viable Product (MVP). Lean product development instructs you to get a working prototype, ship quickly, get feedback, fix it, and do it again.

The simple fact is that it just isn't possible to fully comprehend what good UX means in your particular case until you actually have some dedicated users. Get the basics, communicate your core concept and then get some fingers on the buttons. This is where setting user expectations matters most. If you just need some beta testers, don't promote it as a launch. You actually have more control over user expectations than you think. Many people are willing to try something new and clunky for the coolness factor alone.

However you look at it, you need something out there. And there are many ways to conceptualize your initial product. If you need to brush up, take a look into our free ebook, The Guide to Wireframing — and keep tabs on our Product Design Library for new e-books on prototyping and more.

I. VIABLE BUSINESS

Although there are MVP success stories like Dropbox, which used a bare bones approach to go from 0 to $1 billion in valuation over four years, MVP more often leads directly to a **Minimum Viable Business (MVB)** which builds revenues and traffic more slowly and reliably. The questions you should ask yourself over and over again: can it make money?



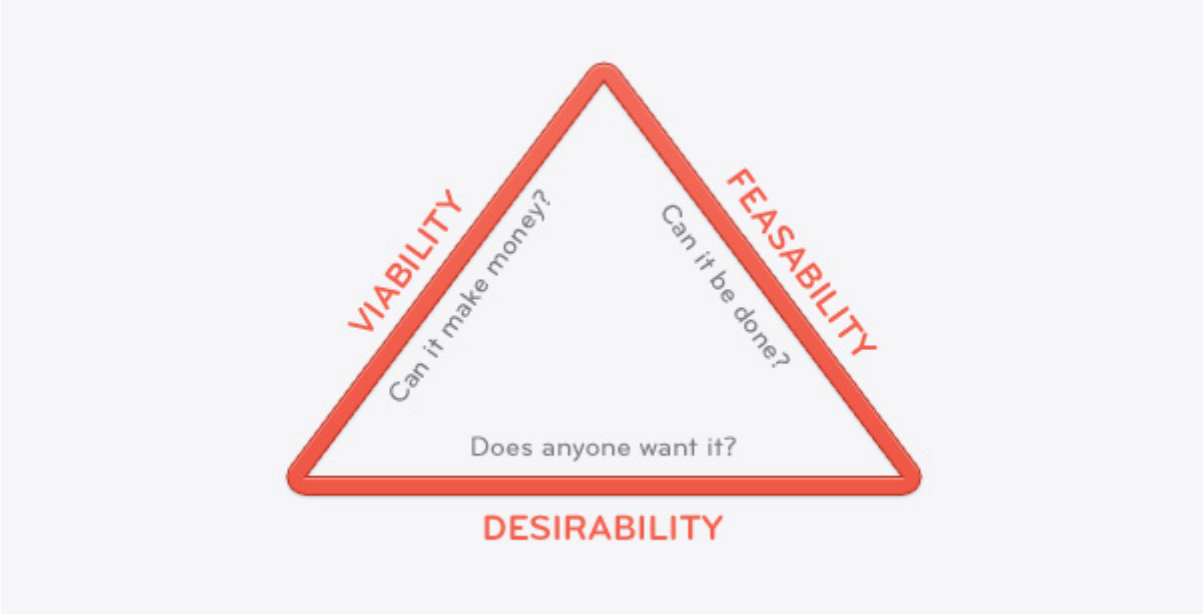
Source: Jon Yongfook, "21 Actionable Growth Hacking Tactics"

II. FEASIBLE PRODUCT

Building a MVB is only part of the story, though. The next step up is making sure your fantasy product can be built with the time, money, resources, and technical capabilities today. It's critical to be honest with yourself here because a lot of people grossly underestimate or flat out ignore this part. You can't really move onto the next step without the product being feasible.

III. DESIRABLE PRODUCT

And, finally, your product has to be desirable, a **Minimum Desirable Product (MDP).** This concept was introduced by **Andrew Chen,** who defined it as "the simplest experience necessary to prove out a high-value, satisfying product experience for users, independent of viability." This guide will help you safely navigate the waters of New Product Development, along with advice on when to focus on UX or Lean and to what degree at each stage of the process.



Source: Intercom, "Asking Questions Beats Giving Advice"

Aarron Walter, Director of UX at MailChimp and author of "Designing for Emotion", echoes a similar sentiment that UX is the key to desirability,

even at the MVP stage. “At first, the primary focus is on usability and learnability. But these days, it’s easy to build an app and get it on the market,” says Aarron. “What’ll make your app stand out from the crowd? Personality, a point of view, an ethos — features can come later, but personality needs to be there from your very first release.”

THE STAGES OF PRODUCT DEVELOPMENT

The most salient factor that determines how much to focus on UX or lean design is the current stage of the development cycle.

One unique aspect of the design process that can be frustrating is that it is path-dependent. Everything is open at the sketching stage, but after that, early decisions put constraints on what is possible later in development. Unless the team is going to scrap everything and start from scratch, which is not always desirable or even possible, additions have to blend in to the existing design. Reduce your uncertainty early for a more successful project all around.

This implies a strategy of reducing the project scope so that all of your resources can be devoted to solving a problem that is fairly well understood. Design simply, get it right, and move on. You are not shooting for perfection but a high-functioning product so you can refine and enhance the core in the next stages of design and development.

Here is a brief breakdown of the three market stages and where your focus needs to be.

I. THE TECHNOLOGY STAGE — LITTLE UX, ALL LEAN

1. **Objective** — Assembling the plan together and figuring out what the market

needs

2. **Key considerations** — Focus on MVP, with an emphasis on the Viable. You can't focus on UX yet is because you don't have any users yet. You know who could use it, but you don't know who will. Get a working prototype together and get it out there to find out who needs it and how they are using it in the real world.
3. **Success criteria** — Consumers outside the beta test want to use it.

II. FEATURE STAGE — LIMITED UX, LESS LEAN

1. **Objective** — Settle on the most important features to develop based on user demand.
2. **Key considerations** — Start molding the UX as part of the decision process to engender specific emotional responses for the user. Do you want them to feel confident, curious or hungry for more? Keep a close eye on what features competitors offer, but don't be in a hurry to get anything out the door until they work properly with the right UX.
3. **Success criteria** — Users comment about how they are using it and how they feel about it.

III. EXPERIENCE STAGE — ALL UX, NO LEAN

1. **Objective** — Researching what users are doing with your product means everything at this point.
2. **Key considération**s — Forget about adding features unless absolutely necessary. What are the biggest pain points customers have with using your product? How much trouble would it be for them to switch to a competitor now?

This is a good time to study winning UX designs from around the world and figure out how to apply these insights to your product. The Hierarchy of Efforts is a great way to structure your UX improvements at this point.

3. **Success criteria** — Unsolicited customer referrals and viral adoption.

At UXPin, our wireframing and prototyping app is admittedly still at Stage II in this process. That said, we have done a lot of customer development and dramatically improved our core product recently to create a more complete and enjoyable experience. Soon we'll be ready for Stage III, optimizing UX, and that will be a really exciting time for us — and our users.

UX DESIGN VS. LEAN DESIGN

The two opposing forces really aren't all that different since the user is in the driver's seat. The sooner you get on board with their needs, the more successful your product will be. Nobody cares that your product does a million things. They only care if it works for them when they open the box.

Here is a quick rundown of what each process looks like for designing a new app.

EXAMPLE OF A UX DESIGN PROCESS

1. Identify people, problem, project — People need an app that turns a week of recipes into a shopping list instantly.
2. Generate lo-fi prototypes — Based on recipe apps and shopper apps, cobble together a shopping engine and try to actually use it at a store
3. Design — This is where UX and creativity really have to get serious.

4. Get Buy-in — Gather stakeholders and measure the excitement. This is the most undervalued stage and the point where products go bad due to lack of backing or not listening to users.

EXAMPLE OF A LEAN DESIGN PROCESS

1. Observe and Brainstorm — People shop by category (dairy, meats, vegetables) and make meals out of what is available. An ingredient focus can change the shopping experience.
2. Minimal Viable Product — Get feedback from a user test group on a bare bones recipe-to-shopping-list converter app.
3. Gather feedback and iterate — Maybe the shopping list isn't the problem, it's finding recipes easily based on lifestyle choices. Find out where the real problem lies and start over from there.

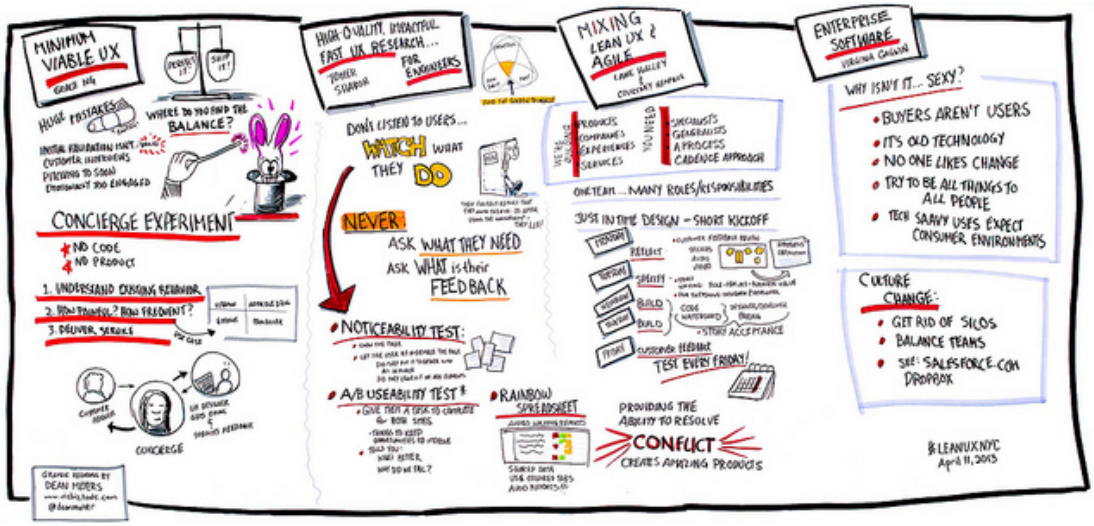
> *"Universal design principles have existed a long time and they are here to stay. If you think your next big app can do without, think again."*

Share Quote

According to Jan Jursa, author of "UX Storytellers" and cofounder of MOBX Conference, the more mature the market in which your product is launching, the more you'll want to favor a UX design process. For him, going MVP is not an invitation to skimp on fundamental product qualities. "Universal design principles have existed a long time and they are here to stay. If you think your next big app can do without, think again," says Jan. "Elements such as harmony and hierarchy are an investment in the perceived experience of every product, as is usability."

STAY LEAN — WITH YOUR EYE ON UX

Lean UX is gaining currency in the design world as companies try to fold the two approaches together seamlessly.



Source: Dean Meyers

The first key to achieving that in the real world is to keep UX as a goal in every stage of design and development, even when the focus is on getting the product into users hands. That way, when UX becomes more important, the design can easily incorporate the necessary changes.

The second key is more feedback from all stakeholders. That doesn't mean try to please everyone, but it does mean that the funding sources and management team are just as valuable a resource as the users when it comes time for implementation.

> *"Launch with the most optimized site that you can improve rather than overhaul. Your site is not carved in stone."*

Share Quote

Frank Lloyd Wright probably said it better than anyone: *"You can use an eraser on the drafting table or a sledge hammer on the construction site."* It is far more effective to use an eraser early because customers don't really appreci-

ate it when you take a sledgehammer to what they see as “their” website. Launch with the most optimized site that you can improve rather than overhaul. Your site is not carved in stone. It should evolve and change based on user needs and the potentials unleashed by new technology.

CHAPTER SIX

TOP 15 WAYS TO TEST MINIMUM VIABLE PRODUCT

Important techniques for startups and established companies alike

TESTING YOUR MVP

Though the MVP provides a means to test hypotheses as a starting point, it does not imply that it is easy to build. The idea behind this exercise is not to see if the product can be built in terms of technical feasibility. Rather, it is to see whether you should be building it in the first place and, more importantly, whether it's solving a problem other people find worth paying for.

Source: Dropbox

Vladimir Blagojevic, founder of Grant Snap & Lean Startup Circle Brussels writes about the importance of building a product that people want to use and pay for. In order to reach that stage, however, you need to make sure your product passes certain tests. Time and money are valuable resources and wasting them on building a product that doesn't meet that criteria is out of the question.

MVP tests are designed not just to answer technical questions about the product, but also to test fundamental business hypotheses about the viability of the market it exists in. For example, a company wants to see if unmanned drones would be valuable for collecting data on the crop health. The project may be deemed

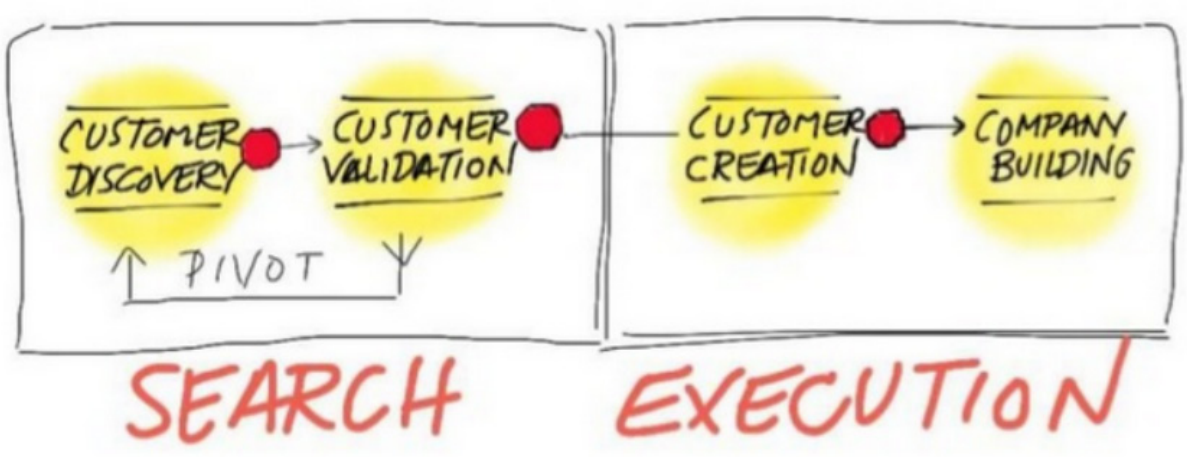
technically viable, but the MVP could ultimately fail because it does not test the viability of business by verifying market and company capabilities.

The complexity of your MVP depends on the type of product you're building, and different kinds of MVPs can range from vague Adwords tests to early prototypes. Once you determine the hypotheses you need to test with your MVP, here are some of the testing techniques you can put to use to get reliable data from actual users and utilize it:

1. Customer Interviews
2. Landing Pages
3. A/B Tests
4. Ad Campaigns
5. Fundraising
6. Explainer Videos
7. Piecemeal MVPs
8. SaaS & PaaS
9. Blogs
10. Manual-First (aka "Wizard of Oz") MVP
11. Concierge MVPs
12. Digital Prototypes
13. Paper Prototypes
14. Single-Feature MVPs
15. Pre-Order Pages

1. CUSTOMER INTERVIEWS

"In a startup no facts exist inside the building, only opinions," says Steve Blank, co-author of The Startup Owner's Manual and creator of the Customer Development Methodology. In his book The Four Steps to the Epiphany, he talks about the Customer Problem Presentation, an important part of the customer validation process that helps you test your hypotheses with actual customers.



Source: Driving Corporate Innovation: Design Thinking vs. Customer Development

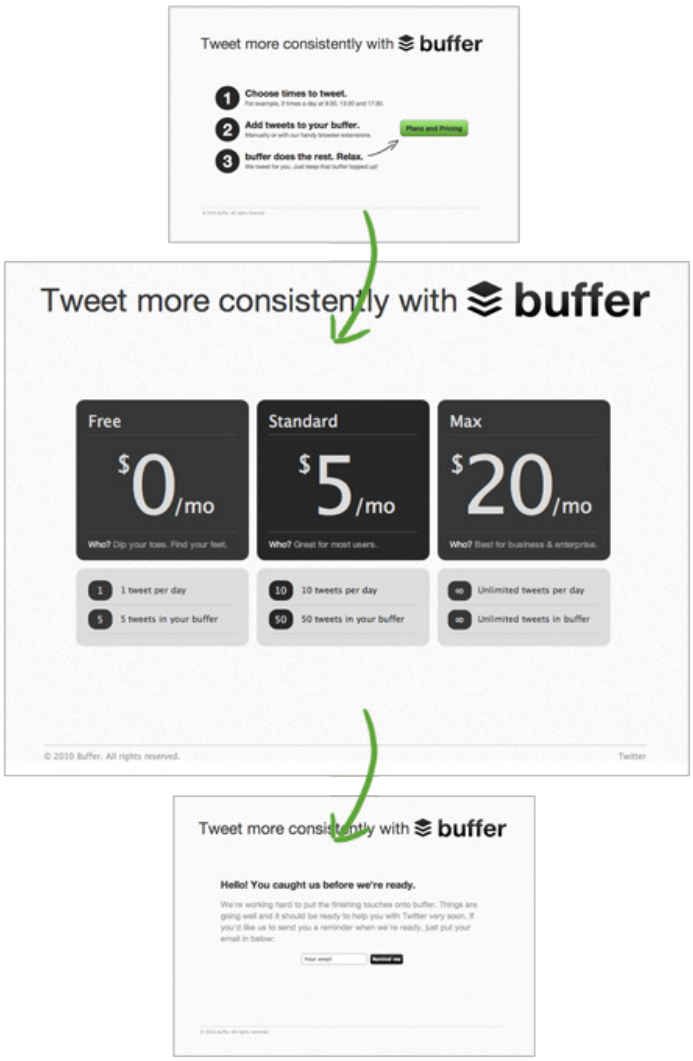
This is essentially an unscripted interview with customers designed to elicit information about the problem your product is trying to solve. These interviews are meant to be exploratory rather than as a sales pitch for your product, functional or otherwise. This process can be continued by listing down the problems you assume your product will solve and then asking what the customer thinks about them as well as how they would rank each problem.

These interviews can be a goldmine of actionable information, because even if your assumed problems turn out to be not as important to the customer, you still have valuable data that can help pivot your offering.

2. LANDING PAGES

The "Landing Page" is the first page visitors and potential customers visit when

they're led down the funnel towards your product. It's a marketing opportunity where you can explain your product's features and have them sign up, but at the same time and for the same reasons, it's also a great MVP that lets you test your product against real-world market expectations.



Source: Idea to Paying Customers in 7 Weeks - How We Did It

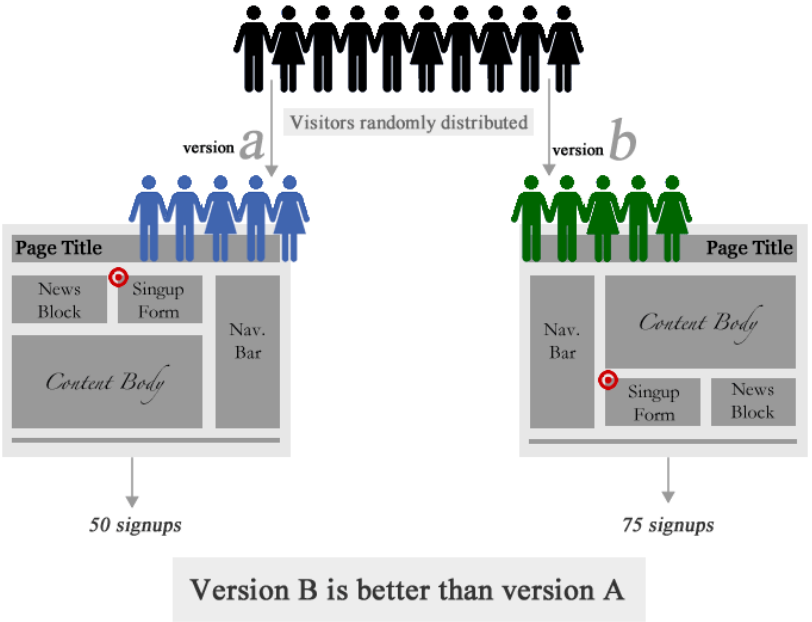
Landing pages are often misused as glorified email capture pages, but they can be used more extensively to test the product. Joel Gascoigne fleshed out the first landing page for **Buffer** and used that to gauge demand for various features and price plans rather than just to build up a mailing list. This was as simple as adding an extra page between the features page and the signup form; the interstitial page showed a pricing table, and the visitors could select the plan that appealed to them. These extra clicks not only showed the visitor's interest in the product,

but also gave the team real-world data on what kind of pricing would be appropriate for the market.

Kate Rutter, instructor at Tradecraft and co-founder of Luxr, is a big fan of using landing pages to "sell first, build later". In order to be most effective, landing pages need to be able to provide the right information to customers in the right context. Remember that the objective is validated learning, so collecting visitor analytics with tools like Google Analytics, KISSmetrics or CrazyEgg is the most important part of it. You also need an effective value proposition and call to action. For even further learning, you can also run A/B tests on the page's content to help nail down what kind of pitch works best for conversion.

3. A/B TESTS

A/B Tests are used to test the effectiveness of any changes to your product or marketing. Various analytics tools can be used to test how visitors react to the design decisions you make, eliminating the guesswork when it comes to improving the product. A/B testing allows you to test two versions of the page or marketing copy and lets visitor interactions determine which one performs best.

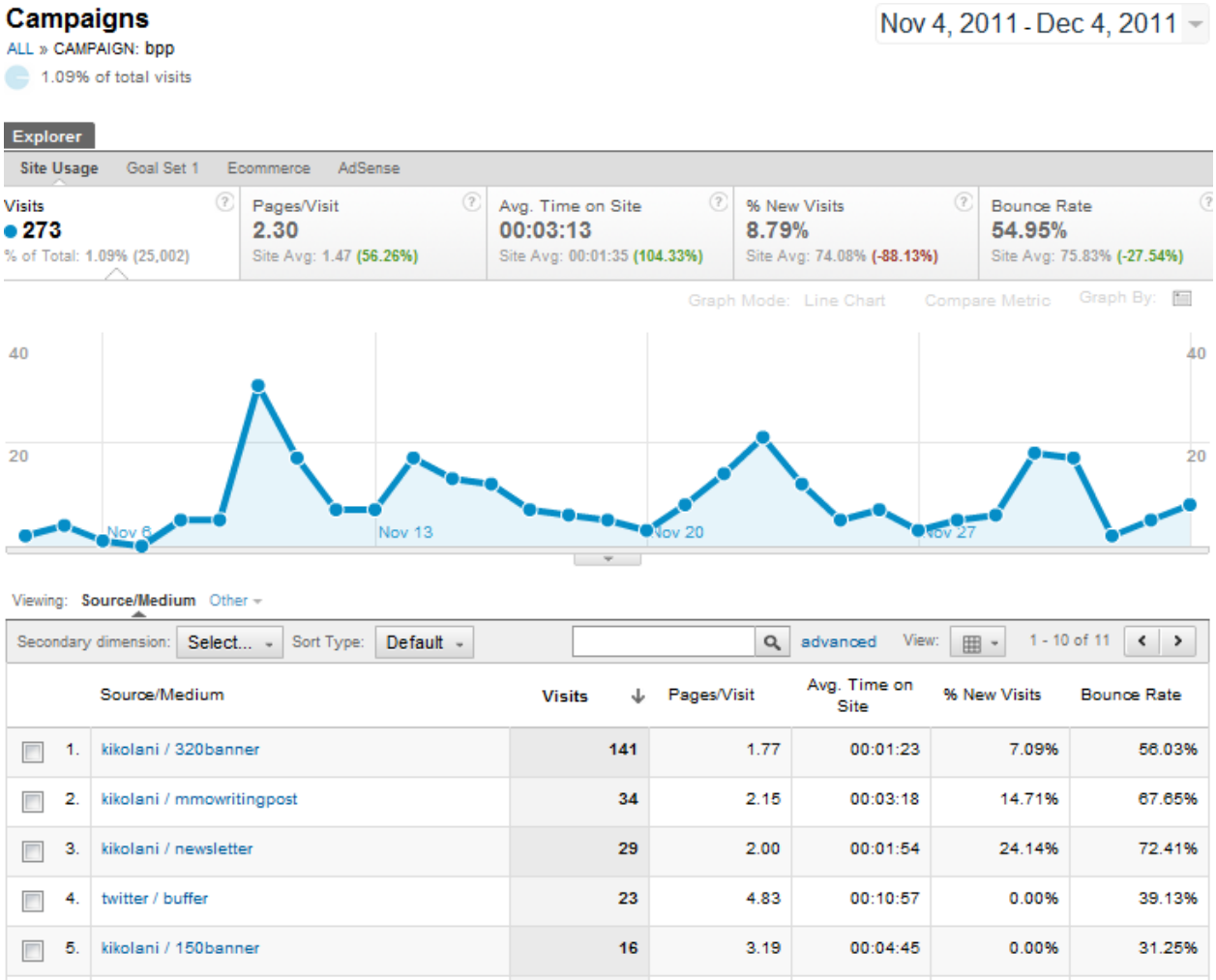


Source: The Ultimate Guide to A/B Testing

A portion of your visitors see version A, while the rest see version B. In the end, using data gathered by analytics tools like Optimizely, Unbounce or Google Analytics, you can measure each version's performance on a set of metrics like bounce rates, conversion or usage.

4. AD CAMPAIGNS

Perhaps counterintuitively, ad campaigns are a great way of running market validation surveys. Google and Facebook are platforms that allow you to drill down demographics to the particular target customer you're trying to reach, and this lets you run a low-fidelity test to see which features or aspects of your product are most appealing to them.

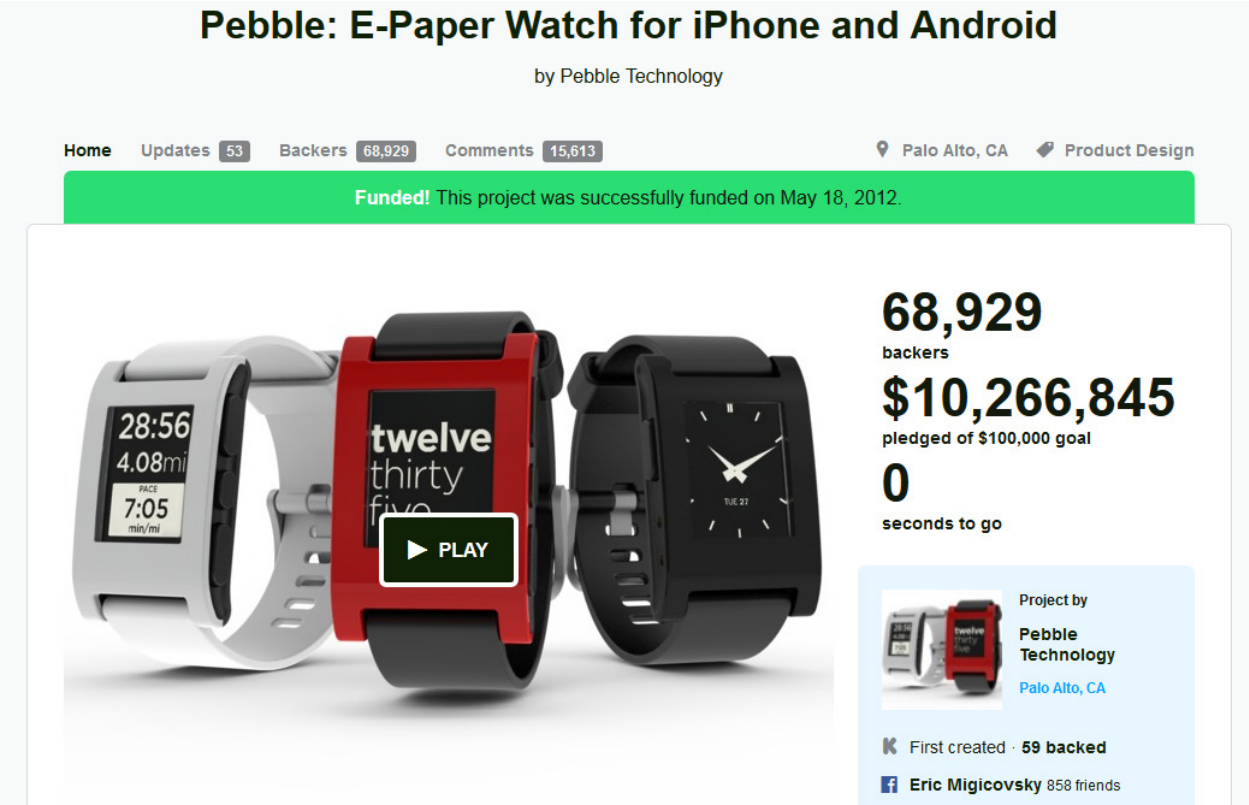


Source: How to Use UTM Parameters

Running a campaign through these services gives you statistics like click-through-rates and conversions which can be valuable information in determining what your product will be and how it will run. These can be combined with A/B tests. Competition in the search marketing space is fierce, so it's important to remember that running an Adwords campaign for your MVP is not going to get you a lot of exposure. But for testing your hypotheses and learning, it's priceless.

5. FUNDRAISING

Crowdfunding websites like Kickstarter and Indiegogo, among others, also provide a great platform for running MVP tests. These websites are essentially collections of MVPs where the market response is judged by the interest people show in the form of contributions to the campaigns. This combines the benefits of validated learning with fundraising for product development and even gives you access to a group of highly interested and actively involved early-adopters who have a stake in the success of your product which is great for building word-of-mouth as well as continuous feedback along the way.



Source: [Pebble, E-Paper Watch for iPhone and Android]

You don't need to look far to hear about success stories that began as Kickstarter campaigns. The **Pebble e-paper smartwatch** and **Ouya gaming console** are perhaps the more popular ones, raising millions of dollars and building buzz even before development began. Of course, some of the hallmark features of campaigns include a compelling narrative, effective explainer videos and useful rewards or incentives for people to back the project.

6. EXPLAINER VIDEOS

If a picture is worth a thousand words, then a video demonstrating your products user experience is worth a million. The most famous example of a startup using an explainer video to validate the market and sell their MVP is **Dropbox**. They began with a 3 minute video that demonstrated Dropbox's intended functionality, which resulted in signups increasing from 5,000 people to 75,000 overnight—all of this in absence of a real product. Of course it also helped that the video

was targeted to tech-savvy early-adopters who appreciated the easter eggs and humorous references in the video.

> *"If a picture is worth a thousand words, then a video demonstrating your product's user experience is worth a million."*

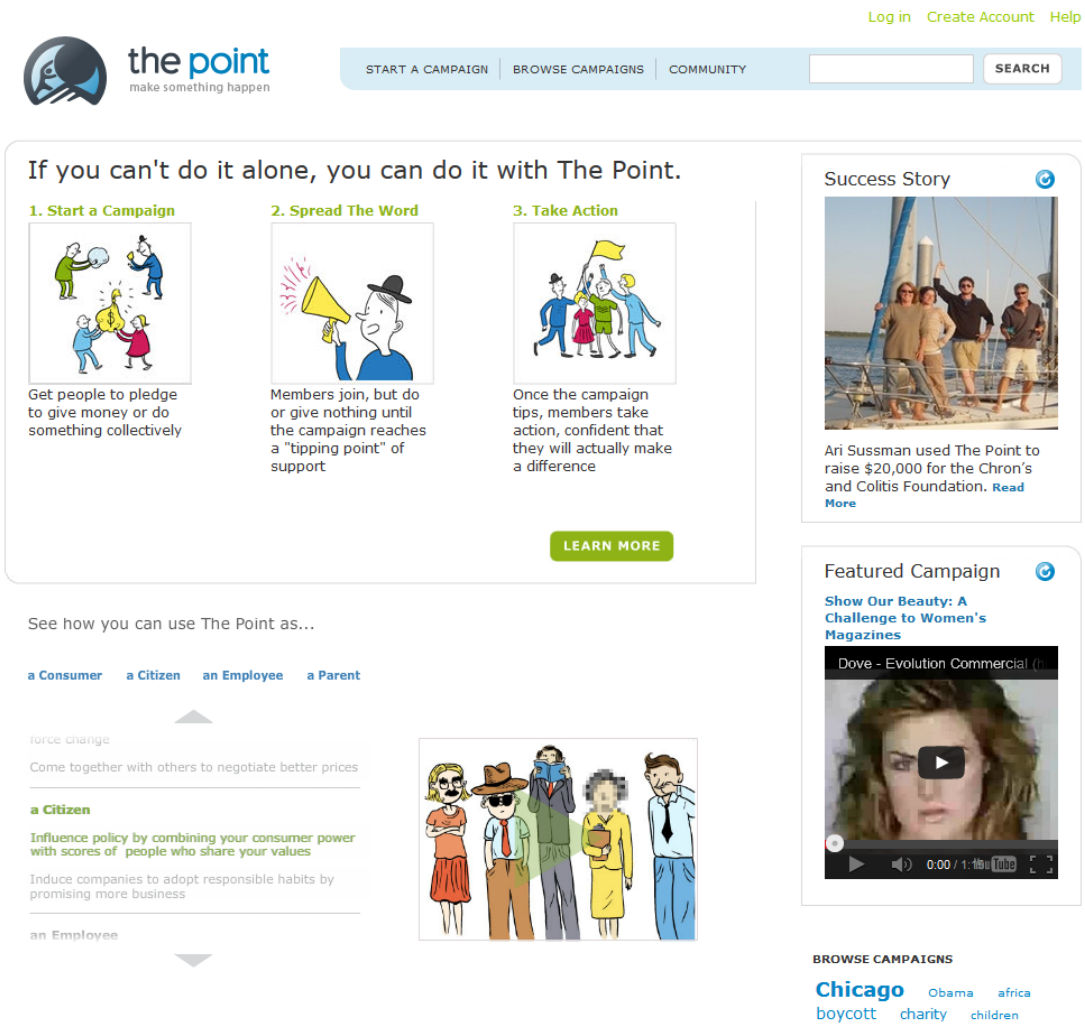
Share Quote

Source: Dropbox

Dropbox's explainer video served as a brilliant validation of the market before the founders ever had to invest in the infrastructure and development needed for a high-tech product like theirs to reach a functional level in the real world. Reaching the target customers is difficult enough, especially when you're designing to solve a problem that many users might not even recognize they have. For Dropbox, perhaps saying they were offering a "seamless file-synchronization app" wouldn't have the same impact. The explainer video instead walks potential customers through what the product is and clearly demonstrates *how* it helps them, eventually leading to *why* they would want to pay for it.

7. PIECEMEAL MVPS

As a mix of Wizard of Oz and Concierge techniques, the Piecemeal MVP means putting together a functioning demo of your product using existing tools and services to deliver the experience instead of building anything yourself. For example Groupon, in its early stages, was a combination of WordPress, Apple Mail and an AppleScript that generated PDFs manually as orders were received from the website.



Source: The Point

Rather than investing time and money into building your own infrastructure, the product can be built using other existing platforms and services as the foundation, effectively using bits and pieces from various sources to make your version of the product.

8. SAAS & PAAS

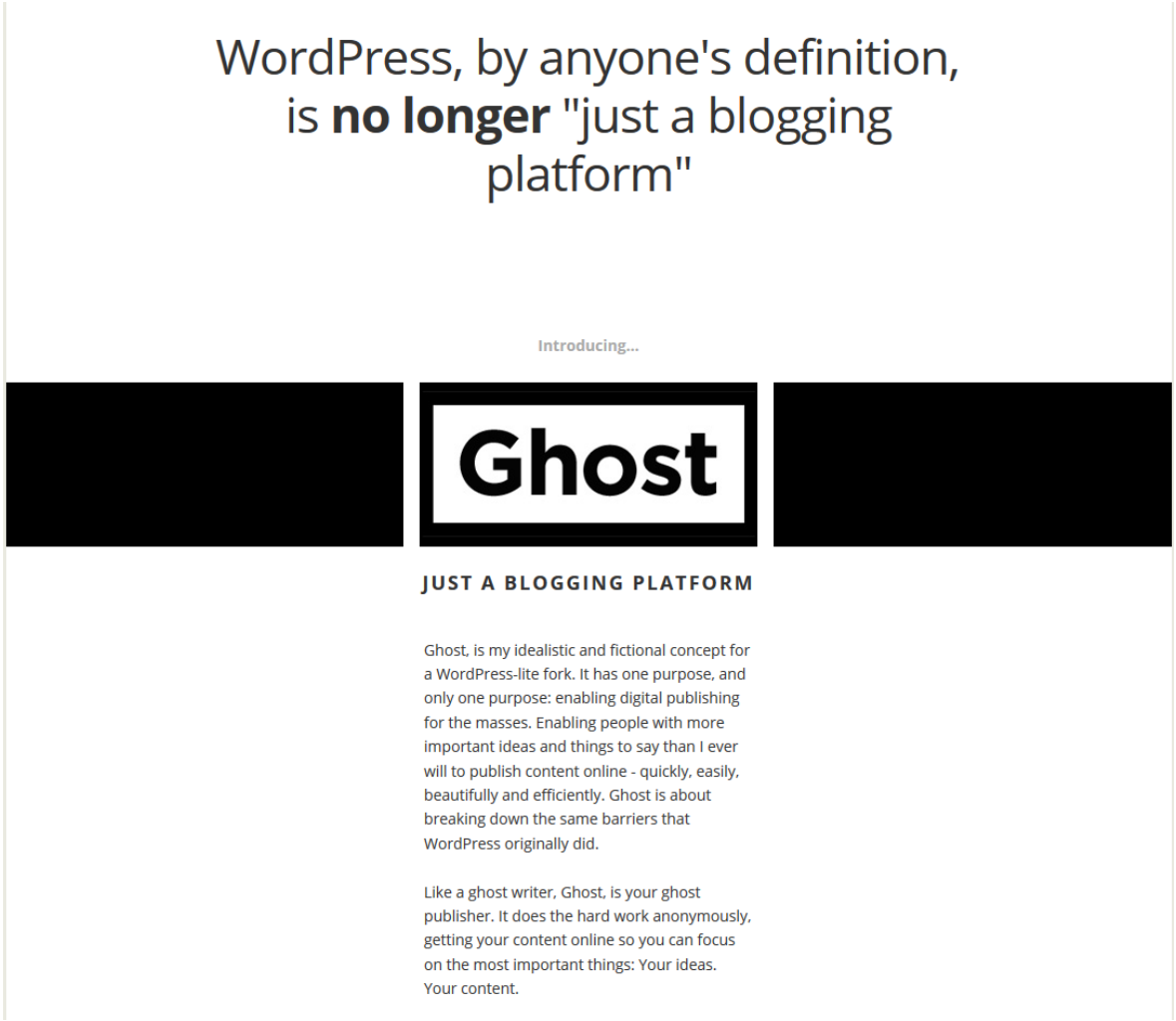
Instead of investing in scalable server technology, relying on cloud platforms like Amazon Web Services, Heroku and MongoDB, Facebook Connect, services like Chargify, Mixpanel, Mailchimp, Google Forms and LiveChat or even platforms like WordPress and Drupal are all great pieces in the jigsaw puzzle that is your MVP test. These services and platforms help you in the development process, speeding up the time it takes to get your MVP to market. Groupon for example, began life as a customized WordPress website where the founders posted deals and manually emailed PDFs to subscribers in the spirit of validating their market potential.

Design and development frameworks can prove to be useful shortcuts for saving time and money. There are multitudes to choose from, some of the most popular ones being Twitter Bootstrap, ZURB Foundation, Ruby on Rails, Django, bootstrap.js or even frameworks like Node.js.

Using a framework or library can significantly speed up your development time. They provide ample documentation and make it extremely easy to get up and running with your MVP. Many of the problems developers face like cross-browser compatibility, mobile-friendly design or code optimization are already taken care of, leaving you free to focus on building your MVP rather than the design or development that is meant to support it.

9. BLOGS

Blogs are a great way of validating ideas with the right target market using minimal effort. **Blogging platform Ghost**, and **App.net** began in concept on their founders' blogs where they continued to flesh out their ideas and gain support from a community of followers and supporters.



Source: Project Ghost

The two-way communication from blogs gives an ideal platform to build momentum and gather customer feedback in the MVP development process. Additionally, blogs can also serve as early prototypes of your product. **Eric Ries, author of The Lean Startup** also began his book as a blog, building an audience and demand before signing any publishing deals. Similarly, so did 50 Shades of Grey!

10. MANUAL-FIRST (AKA "WIZARD OF OZ") MVP

Rather than building a video or coding a framework, an alternate option for the initial stages of market validation is to deliver the product or service manually. The "Wizard of Oz" MVP comes from the idea of putting on the impression of full functionality, essentially faking it until you make it. Customers believe they are experiencing the actual product, but in reality the work behind the scenes is being done manually.

60	"Lior	04-Oct Mon	12:00 PM	X	Roast Turkey - SSB Catering	15			
		Billed					38 K	16-Sep T	
60	"Lior	01-Oct Fri	12:00 PM	X	Burger Bar - Brickhouse	15			
	"Lior	30-Sep Thu	12:00 PM	X	Sandwich Lunch To-Go	18	38 K	03-Sep F	
60	"Lior	29-Sep Wed	12:00 PM	X	Bah Mai - Dinosaur	15			
60	"Lior	28-Sep Tue	12:00 PM	X	Burmese - Mandalay	15	38 K	29-Jul Th	
	"Lior	27-Sep Mon	12:00 PM	X	Pasta - Pasta Paradiso	15		Billed	
60							38 K	22-Jul Th	
	"Lior	24-Sep Fri	12:00 PM	X	Indian - Mehfil	15			
60	"Lior	23-Sep Thu	12:00 PM	X	Chicken Piccata - Pasta Paradiso	15	38 K	15-Jul Th	
	"Lior	22-Sep Wed	7:00 PM	X	Miso Glazed Salmon - Pasta Paradiso	12			
60	"Lior	22-Sep Wed	12:00 PM	X	Assorted Sandwiches - Deliboard	15	38 K	08-Jul Th	
	"Lior	21-Sep Tue	12:00 PM	X	Sausages - Rosamunde	15			
60	"Lior	20-Sep Mon	12:00 PM	X	Roast Turkey - SSB Catering	15	38 K	01-Jul Th	
								Billed	
56	"Lior	17-Sep Fri	12:00 PM	X	Vietnamese - Jasmine Garden	15	38 K	24-Jun T	
60	"Lior	16-Sep Thu	12:00 PM	X	Burger Bar - Brickhouse	17			
	"Lior	15-Sep Wed	12:00 PM	X	Korean - Stone Korean Kitchen	15	38 K	17-Jun T	
60	"Lior	14-Sep Tue	12:00 PM	X	Chinese - Jasmine Tea House	15			
	"Lior	13-Sep Mon	12:00 PM	X	Smoked Beef Brisket Brasola, Stuffed	15	38 K	10-Jun T	

Source: How I Started ZeroCater

Arram Sabeti, founder of ZeroCater started with a giant spreadsheet which he used to keep track of companies and caterers he could connect with. **Zappos began the same way,** with its founder Nick Swinmurn putting up photos of

shoes from local shoe stores on a website to gauge demand for an online store. When someone ordered the shoe online, he would come back to the store and buy it. Instead of first investing in infrastructure and inventory, this gave Zappos a chance to answer the question of whether their product would be accepted by the market.

This approach also allow for greater interaction with customers at this crucial stage when you're designing the product. Observing actual customer first-hand is always more useful than a hypothetical customer survey, and it's the fastest way to discover whether it's solving a real customer problem. The expedited learning, albeit at a small-scale, provides opportunities to test many assumptions you might be making about the product or the marketplace. By doing it all manually, you give yourself the chance to try different things on-the-fly to see how customers react before you scale. To the customer of course, the product works and the behind-the-scenes work doesn't matter.

These MVPs undoubtedly require a significant effort, but can be ultimately worth it for the focus they provide on the problem rather than the solution. Zappos for example, eventually grew into one of the most successful online businesses and was acquired by Amazon for $1.2 billion in 2009.

11. CONCIERGE MVPS

The Concierge test is similar to the Wizard of Oz MVP, except instead of faking a working product, you're upfront about the manual work and the product or service is delivered as a highly customized service to selected customers.

Source: Defining your vision with a minimum viable product

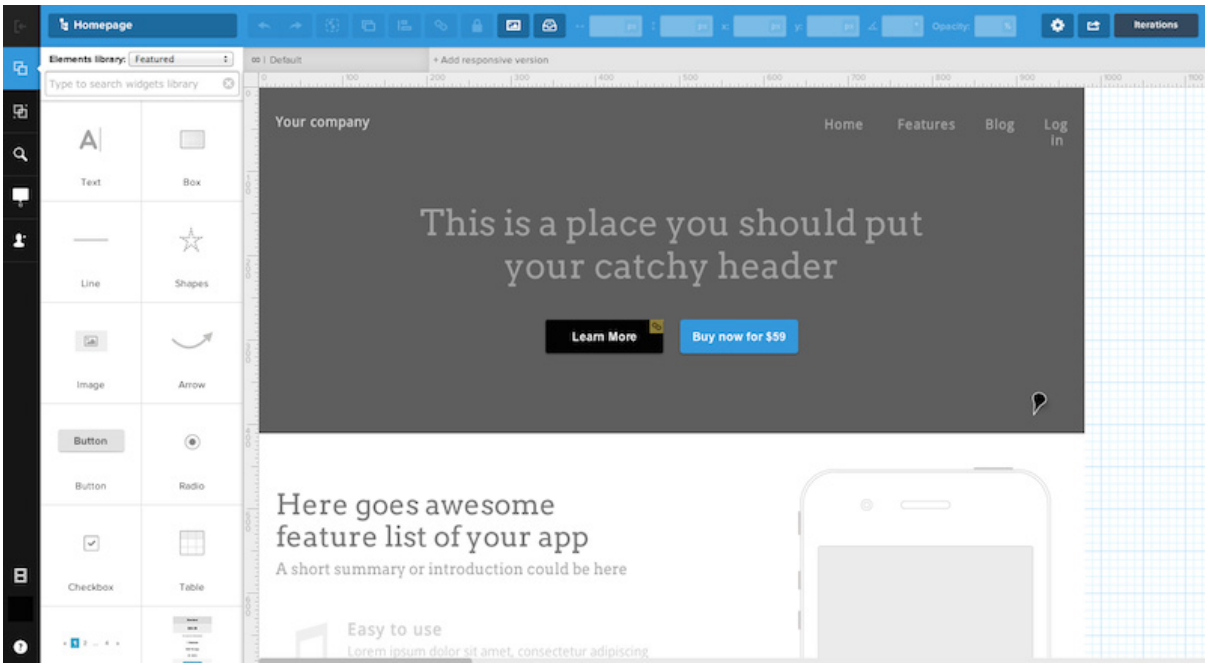
Rent the Runway tested its online dress rental business model by providing an in-person service to female college students where anyone could try the dress on before renting them. This validated their riskiest hypothesis that women would rent dresses and served as a great concierge MVP that put the business in front of customers and got them feedback.

Time is valuable, especially at this stage and and running through the process manually also reveals other aspects of the customer experience that can prove valuable further on. Instead of putting in resources towards building a real product, these MVP tests can answer the more important question first: are you building something that customers will use and pay for?

12. DIGITAL PROTOTYPES

Mockups, wireframes and prototypes can be used to demonstrate the product's

functionality in a way that mimics the actual usage. These prototypes MVPs can range from low-fidelity sketches to screenshot previews to more complicated "dummy" applications that demo the user experience.



Source: UXPin

You can use collaborative wireframing and prototyping tools like UXPin that let you express what you want to build and share those ideas transparently with the team.

13. PAPER PROTOTYPES

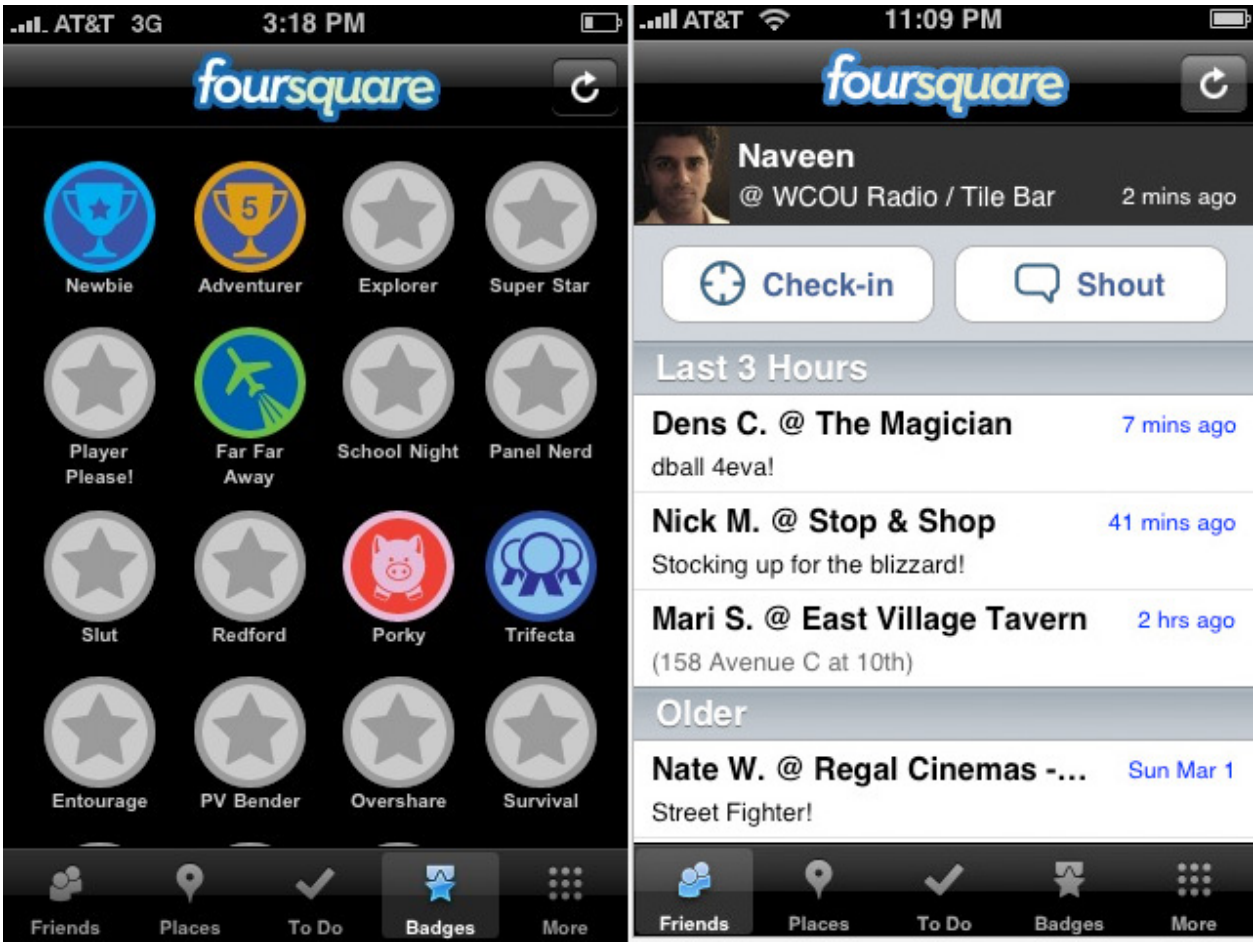
Similar to Digital Prototypes, except these are physical, either made of cutouts or even sketched on paper to demonstrate your product and its user experience. The advantage with paper prototypes for MVP testing is that they can be used by anyone on the team and require very little explaining because it hands you an actual representation of the product.

Source: Caryn Vainio, Winnie Chang, Adrian Kosmaczewski

For physical product development like phones or chairs etc. this technique is invaluable. We cover digital and paper prototyping in greater detail in our Guide to Wireframing e-Book.

14. SINGLE-FEATURE MVPS

Oftentimes it may be best to focus on a single feature of your minimum viable product to save development time as well as prevent users from being distracted with what the product is primarily supposed to be.



Source: [Foursquare scores despite its flaws]

Foursquare, for example, began with the simple idea of letting users check-in to the social network with their location. The first versions of their app reflected this simplicity. Buffer started out with just Twitter support and only one account per user. These restrictions help you narrow down the initial customers and focus on the more important problems, like testing product and market viability rather than worrying about adding support for other mobile platforms.

15. PRE-ORDER PAGES

Similar to the fundraising MVP, the Pre-Order Page MVP lets you present your product to potential customers with the aim of enticing them enough to pay for it before you even build it.

Source: Oculus Rift

Oculus Rift, the virtual reality gaming kit, launched a pre-order page for its development kit before they began production. A lot of projects on Kickstarter begin as pre-orders. This can show you how much demand exists for the product you're trying to build, giving you an indication of whether you should continue or scrap the project. The problem with offering a pre-order is of course that customers might be wary of the possibility that you will not deliver with the product you promise. Nobody likes vaporware, and users who back a project in such early stages demand a return on their faith in you, and of course, on their money.

GETTING OUT THE DOOR

In some ways, building a MVP actually creates additional work because this process of iteration and validated learning requires significant investment of time

and energy. That's why it's important not to get bogged down by unnecessary details and overhead when building MVPs. Eventually, the objective is to figure out if the effort you're putting in is worth it, and you don't want to spend time working on something users don't find useful or want to pay for.

It's also important to realize that when testing your hypotheses, you might want to consider using multiple MVP testing techniques. The one that fits your business model and market best will undoubtedly vary. But the important lesson to take away here is to go out and build it. Think about the biggest assumption your product is making and build an MVP that tests that hypothesis in the market.

CHAPTER SEVEN

BUILDING MINIMUM VIABLE PRODUCTS AT SPOTIFY

Design & development cycle insights from a top-tier product team

Many companies face the paradox of wanting to build a delightful product without knowing if people actually want the product until it's released.

Source: How Spotify Stays Lean

Spotify's vision was to give people the right music at the right time while incentivizing artists by paying them based on number of shares their music received. A tall order, no doubt, when you consider how hard it is to build out such a platform, let alone make it profitable. Yet Spotify defied the odds and grew from zero to over 1 million ***paying*** subscribers in the US — a market foreign to Spotify's native Swedish team and one already teeming with competitors.

So how did they develop a product that fulfilled their vision without driving them into bankruptcy? They took an iterative approach combining elements from Lean Startup, Agile, and MVP methodologies.

To achieve its current 10 million paying and 24+ million total user base, Spotify set out a basic roadmap of prototyping early and cheaply, launching only when a baseline of quality was met, and then iterating based on user feedback. In this piece, I'll explain the goals of each stage and how they all contributed to a sustainable product development cycle.

LEAN & AGILE AT SPOTIFY

According to Henrik Kniberg, Agile and Lean Startup consultant and author, Spotify uses a 4-stage iterative product cycle (Think It, Build It, Ship It, Tweak It) that emphasizes small teams (referred to as 'squads') completing small batches of work while producing a complete product. Let's take a look at the goals of each stage:

- **Think It** — Decide what product to build, then build prototypes and test viability internally.
- **Build It** — Create a physical MVP ready for user testing.
- **Ship It** — Gradually release the MVP to all users while collecting data and improving.
- **Tweak It** — Iterate continuously based on feedback until product is shut down or revamped (returning us to Think It).

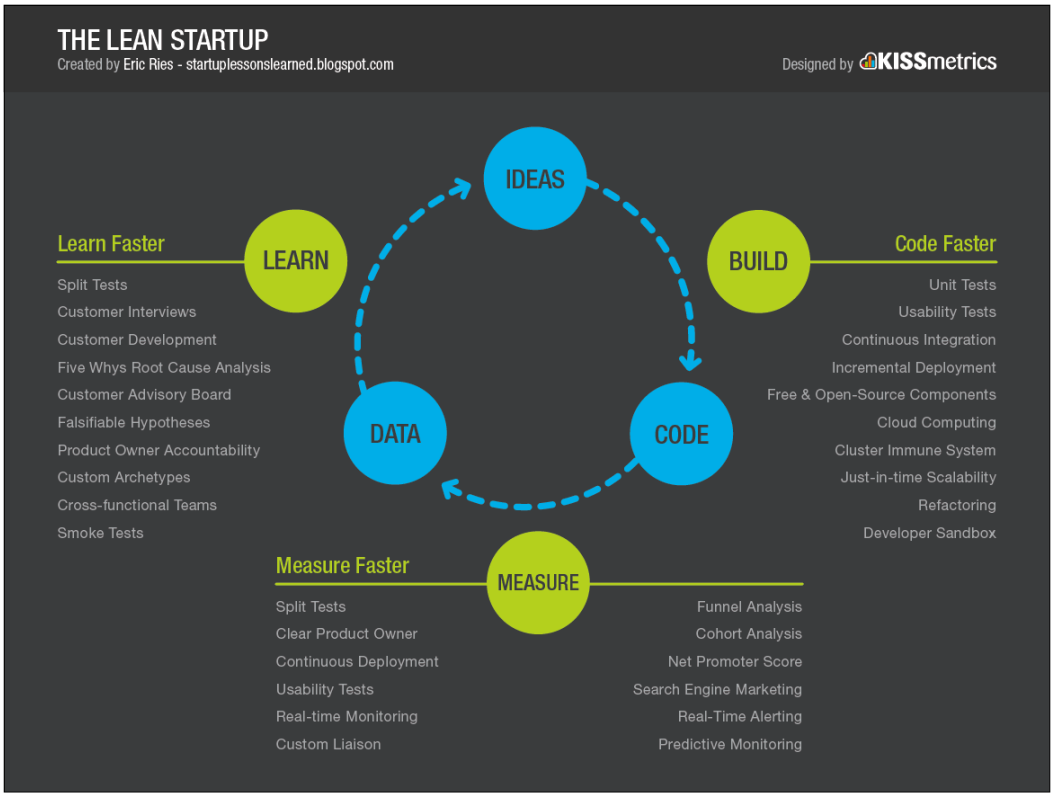
Sound familiar? It should. It's essentially a hybrid process combining practices from Lean Startup and the Agile Methodology.

PRINCIPLES OF LEAN

Lean Startup uses a "Build-Measure-Learn" cycle to reduce waste while achieving quality, speed, and customer alignment. At the heart of Lean Startup is the MVP, a quickly and cheaply produced quality product for learning purposes. Therefore, Lean Startup eliminates the idea that a team can build what it "knows" it will need in the future.

To that degree, each one of Spotify's four stages are Lean since small teams are always working smartly to test assumptions. The "Think It" stage tests the merit

of conceptual MVPs while the "Build It" stage releases a physical MVP only after it's been tested for quality. The "Ship It" and "Tweak It" phases ensure long-term quality and customer alignment by releasing the MVP gradually, learning from feedback, and iterating tirelessly. Spotify does deviate slightly from Lean, however, since the "Think It" stage only tests prototypes internally — Lean emphasizes customer testing as often as possible.



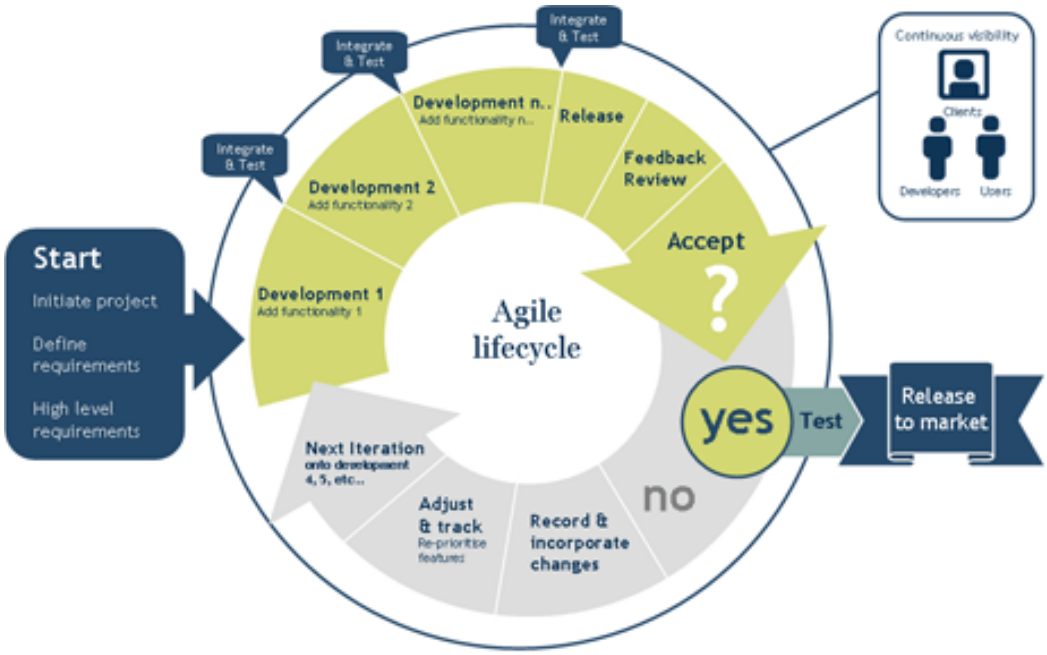
Source: Entrepreneurs Love to Learn

PRINCIPLES OF AGILE

Lean thinking is necessary in order to develop the mentality needed for Agile practices.
While Lean is used to efficiently define and build a marketable product, Agile is the means to accomplish this in software development.

Team members from all disciplines collaborate on short bursts of work (sprints lasting 1-4 weeks), and in doing so, react better to requirements changes. The

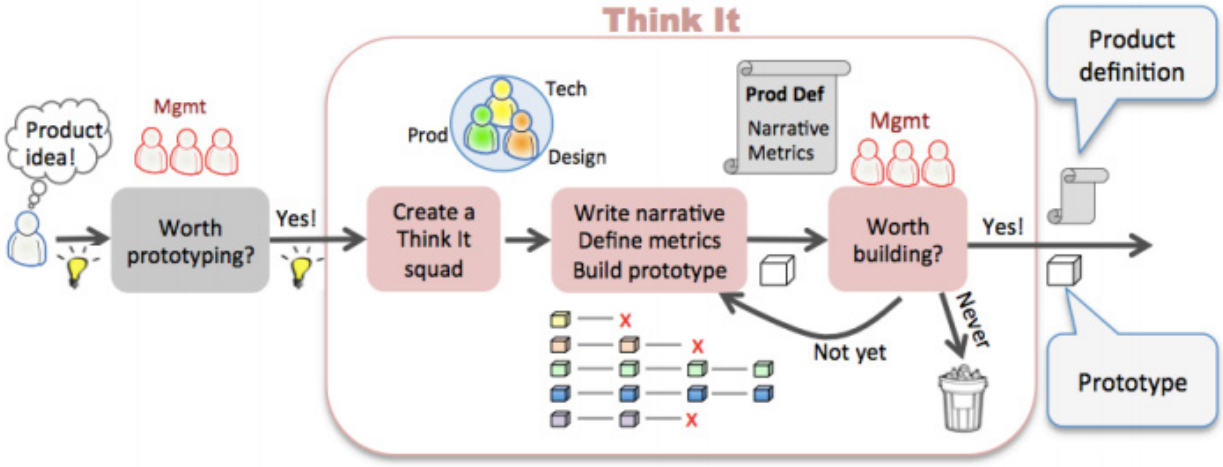
Spotify strategy stresses this collaboration by preserving the same product team throughout all four stages. These sprints are especially important for keeping resources in check during the “Build It”, “Ship It”, and “Tweak It” phases when all the heavy lifting is done. All the testing and validating in each phase also keeps Spotify on the Lean path even if product requirements must change to reflect customer and market needs.



Source: [Agile Methodology]

I. THINK IT

Before committing resources to a project, companies need to evaluate the viability of the idea. The “Think It” stage consists of small teams asking themselves “Why?” rather than just “How?” At Spotify, this phase is the conceptual stage of the MVP since viability is assessed and minimalist solutions like landing pages and prototypes are used to test user demand.



Source: How Spotify Builds Products

According to **Christina Wodtke**, **former General Manager at Zynga**, the importance of looking at viability through a business lens can't be stressed enough. For many companies, it can be tempting to build the perfect viable product simply because they have the resources to do so. By prioritizing viability, Spotify ensures it won't even waste time and money assigning an MVP team if the overall idea isn't profitable.

If, however, management verifies an idea is viable, a small "Think It" team consisting of a developer, designer, and product manager is formed. At this point, the team works on creating a product definition document so that a usable prototype can be built. Using the document, they'll seek to answer questions such as:

- Who will benefit from this and how?
- What are the key metrics that we expect this product to improve? (e.g. songs streamed, number of downloads, etc.)
- What are the hypotheses?
- How will we know if this product is successful?
- Is this a "step change" (a product yielding at least a 2x improvement on the

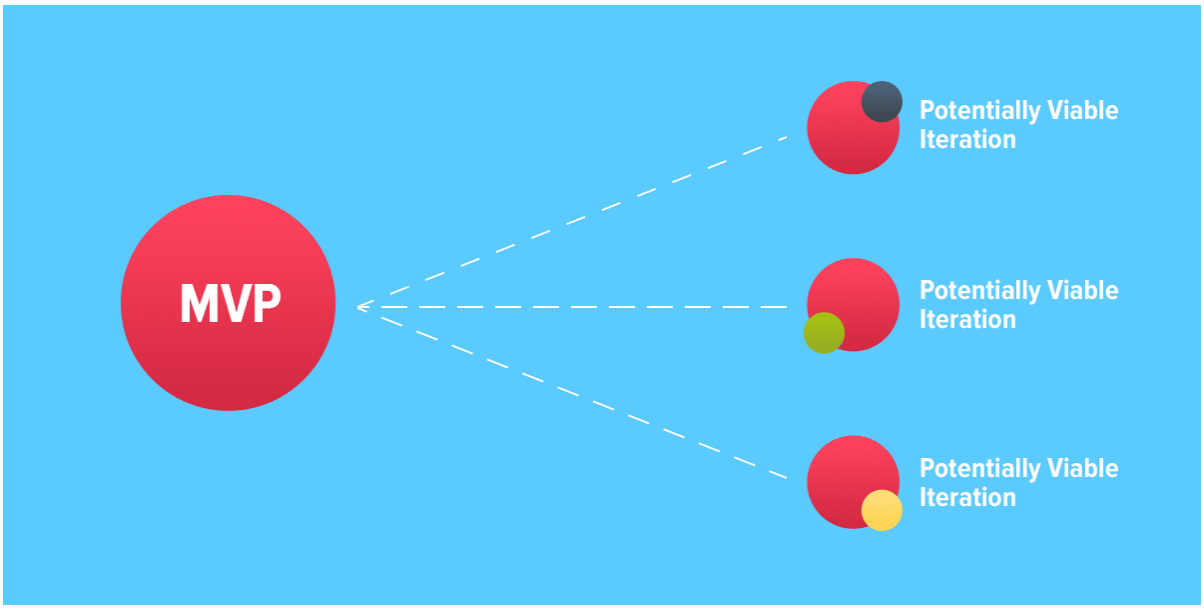
chosen metric)? If only minor metrics improvements are expected, another strong strategic reason should exist.

The goal here isn't to outline features or technical requirements but to create a data-focused value proposition. The heart of the product definition is narrative. And the story told by the product is what the first iteration of the MVP will test.

Matchist.com cofounder, Stella Fayman, aptly states the goal of an MVP is to prove that people will use your product. Landing pages and paid ads are a great low-cost way to gauge interest because you test the basic value proposition first (or narrative, in the case of Spotify) before sinking money into anything. This is precisely what Spotify does. When developing its Mobile Free Radio (one version being "Radio you can save"), Spotify ran a Google Adwords campaign to test narratives. In doing so, Spotify exemplifies applying minimalism towards an MVP.

Once the messaging is finalized through testing, the Think It team builds low-fidelity paper prototypes and high-fidelity runnable prototypes (with fake data). Internal user testing provides feedback on which prototypes best convey the narrative until the list is narrowed down to just a few contenders.

David Aycan, Design Director at the esteemed design and consultancy firm IDEO, explains that multiple prototyping avoids tunnel vision. Ideating on different user experiences puts your eggs in different baskets (preventing over-attachment) and finds the best solution through real data instead of trying to predict what users want. Testing multiple prototypes lets Spotify find the most viable MVP by focusing on breadth rather than depth.



Source: *Using Minimum Viable Products*

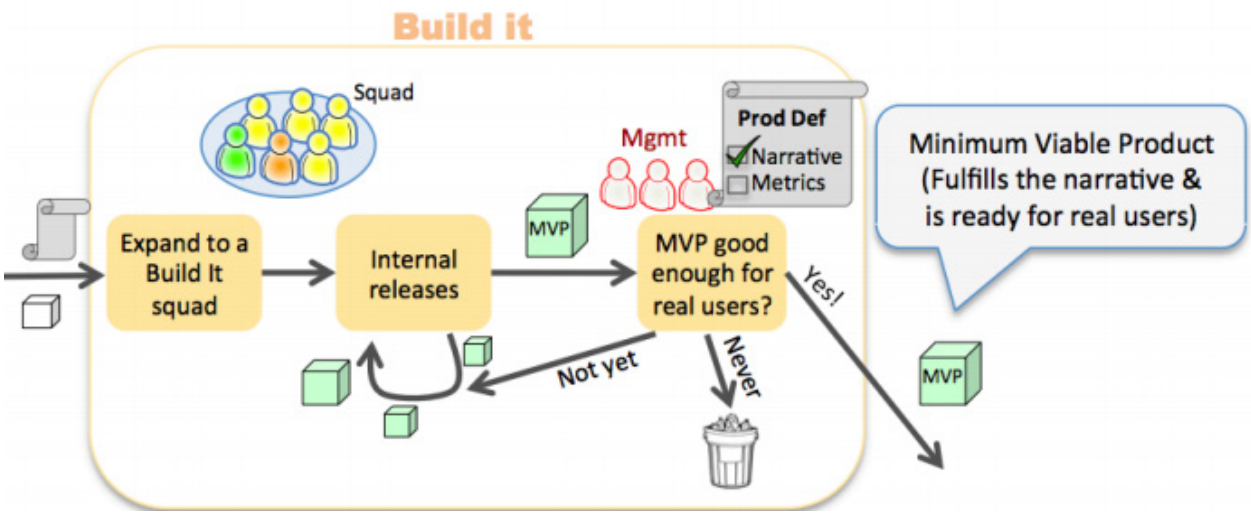
In UXPin, we can turn wireframes into prototypes rather quickly in our web app. We take a similar iterative approach to Spotify by starting out with several lightweight, even low-fidelity, prototypes and narrowing down the options from there. However, we like to involve a small subset of friends and customers in the prototype testing (unlike Spotify which just keeps it internal) since they help us "think outside the building." Ultimately this keeps our team centered on a good customer solution instead of just a technical marvel.

Spotify, on the other hand, will only move forward if it can match the right prototype to the narrative. Because it only involves prototyping and experimenting, the Think It stage is the essence of MVP thinking — the team fails quickly and cheaply, and keeps learning until they find the exact product to build.

II. BUILD IT

Now that a product is decided, the team moves beyond testing concepts to cre-

ating a physical MVP that is good enough to release to external users and test assumptions.



Source: How Spotify Builds Products

The right balance of minimalism and quality must be struck with the physical MVP. Building a feature-complete product requires too much time and money, but rushing a feature-poor product out the door would embarrass Spotify and yield no useful learnings. As such, the team must create the smallest possible thing of quality that still fulfills the narrative and delights users.

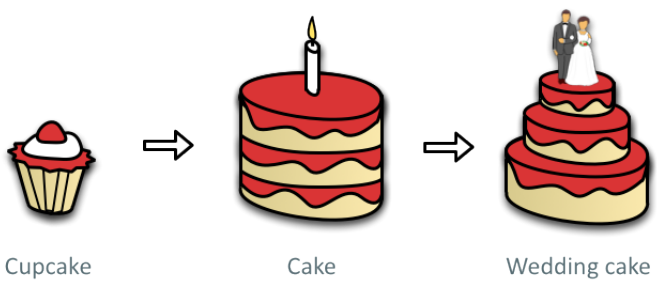
As **Former Apple chief evangelist, Guy Kawasaki,** asserts in his MVP philosophy, the physical MVP doesn't need to be perfect but it must be revolutionary. Early adopters are incredible force multipliers when it comes to early-stage products, and the only way to gain their influence is to create an MVP that embodies five important qualities — it must be:

1. **Deep** — Great products have just the right level of functionality and don't become useless after just a few weeks.
2. **Intelligent** — Great products map specific solutions to pain points (and make customers aware of problems they didn't even know they had).
3. **Complete** — Great products are completely usable, even in early stages.

4. **Empowering** — Great products incite users to action and encourages them to spread the news to help others.
5. **Elegant** — Great products have intuitive user interfaces and work the way people think they should.

You'll see in the above diagram that the key question Spotify's product and management team asks is "Is the MVP good enough for real users?" By making its MVP narrative-complete and not feature-complete, Spotify is able to inherently satisfy all five qualities for a desirable and usable MVP. ***Perhaps, a better term for Spotify's MVP would be MLP (Minimum Loveable Product).***

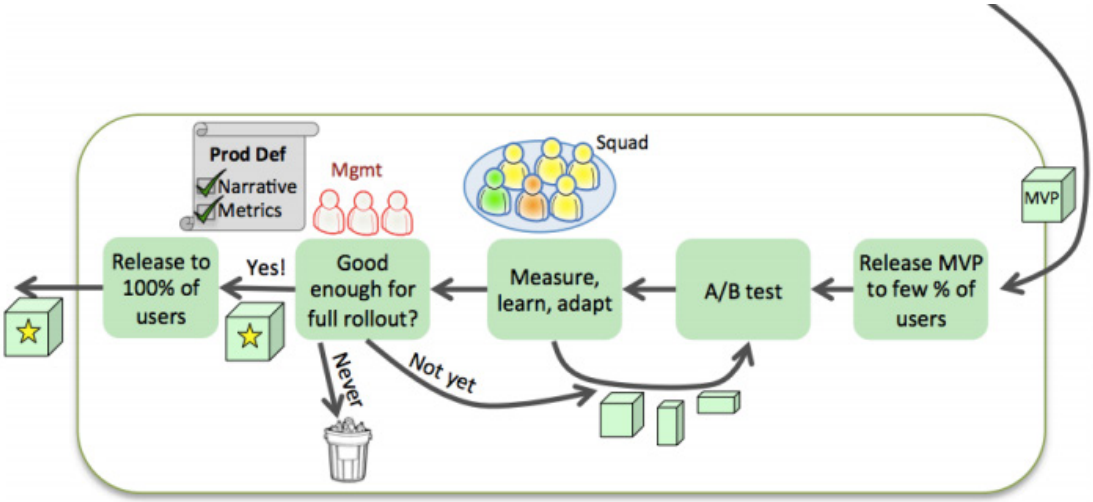
The Cupcake Model, which was first coined by **Brandon Schauer, CEO of Adaptive Path**, emphasizes desirability and completeness regardless of iteration. The analogy states that instead of starting with an uninteresting cake and then adding filling and icing, you start with a cupcake and iterate it into a cake. That way, you invest less resources and the MVP is desirable — people will pay for a cupcake because it's complete with filling and icing. On the other hand, a dry cake requires more resources to create and is unlikely to be good enough for real users. By following a Cupcake Model, Spotify more importantly avoids misleading conclusions: "Well, we baked a plain cake and nobody wanted it, so the cake failed and we shouldn't bother adding frosting or filling."



Source: The Value in a Cupcake

III. SHIP IT

The purpose of the Ship It stage is to gradually roll out the product to all users while measuring and ensuring that the product fulfills its promise in the real market.



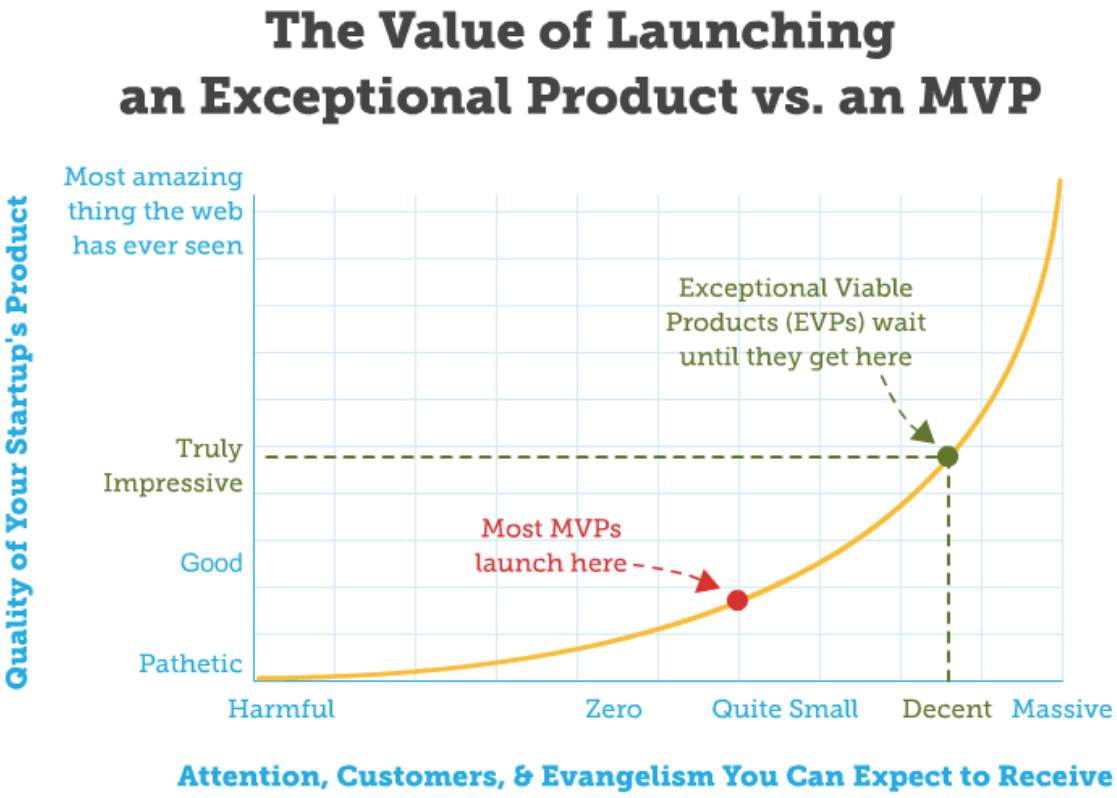
Source: How Spotify Builds Products

Spotify starts by releasing to a small percentage of all users (usually 1-5%) to collect early feedback. During this stage, the hypotheses that were internally tested during the Think It stage are now externally validated.

As you'll see in the above illustration, the beauty of this stage is that Spotify doesn't need to get it right on the first try. Collecting data, iterating the MVP, and then A/B testing the changes allows for continuous rounds of maximized learning at minimized cost. Releasing an MVP early and to a small user base allows Spotify to iterate until it becomes an EVP (Exceptional Viable Product) that is ready for all users.

First coined by **Rand Fishkin**, **co-founder of Moz**, the EVP prevents companies from prematurely releasing an MVP that just isn't ready to excite early adopters. Spotify understands that first impressions matter a lot, so it takes a cautious

approach of having a limited release of something good before making it great and unleashing their brilliance. Based on the below diagram, Spotify releases its MVP somewhere between "Good" and "Truly Impressive" and then uses customer feedback to improve it into an EVP.



Source: [Moz, "7 unlikely recommendations for startups & entrepreneurs"]

When management and the product team agree that the product is having the intended impact on the small user group (based on the product definition), Spotify will gradually roll it out to all users, while still measuring and improving.

IV. TWEAK IT

At Spotify, this stage is the longest, and perhaps most important. Unless products get scrapped during the previous MVP stages, they spend most of their life in this completely iterative phase.

Source: How Spotify Builds Products

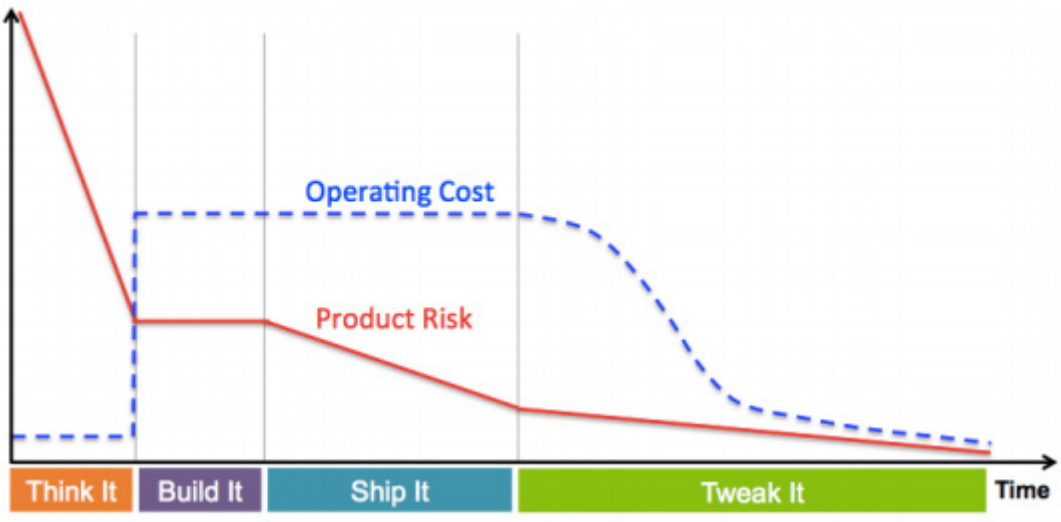
While they may have proven themselves to a certain extent in the Ship It stage, Spotify's products are never considered feature-complete. The team continues the Ship It process of gathering customer feedback, experimenting, and A/B testing to improve the product, resulting in either major rework or just minor tweaks. But, at some point soon, they may reach a point of diminishing returns when the cost versus benefits of new features just looks less and less attractive.

As demonstrated in this article on value vs. complexity, this is an important crossroad where feature prioritization is required. While Spotify believes that no product is truly complete, it also understands the danger of feature creep. Once a product hits its "local maximum" where small tweaks won't really improve things, Spotify's product team and management evaluates if being at the top of the hill is sufficient, or if a higher peak is in sight. If the effort isn't worth the time, the team will move on to other products. Otherwise, the product returns to the "Think It" stage so it can be reworked and leap to the next peak of quality. Spotify's Tweak It stage ensures that it does not fall victim to the idea that first to market will always stay king of the hill.

Gerard J. Tellis, **a professor at USC Business School**, believes that product quality has become so important in recent years that network effects alone will no longer protect companies who are first-to-market. In fact, network effects actually reinforces competition for quality by driving customers to superior products. According to Tellis, the average duration for market leadership in the software industry was only about 3.8 years. When you consider that Spotify is slowly inching towards iTune's market share as of 2014, Spotify's evolutionary product strategy is definitely working.

MORE PRODUCT STAGES = LESS COST, LESS RISK

For many companies, one of the most dangerous mistakes is building the wrong product. They sink enormous cost into ideas that they think customers want and then speed down the path of no return.



Source: How Spotify Builds Products

As you've seen, Spotify's 4-stage product cycle helps them carefully find the right product early and build it quickly and sufficiently. The length of each stage may differ, but the constant balance between minimizing resources and maximizing

product quality is consistent.

> *"Don't be afraid to fail fast and quickly — as long as you keep testing, your most profitable idea might just come from what you learn."*

Share Quote

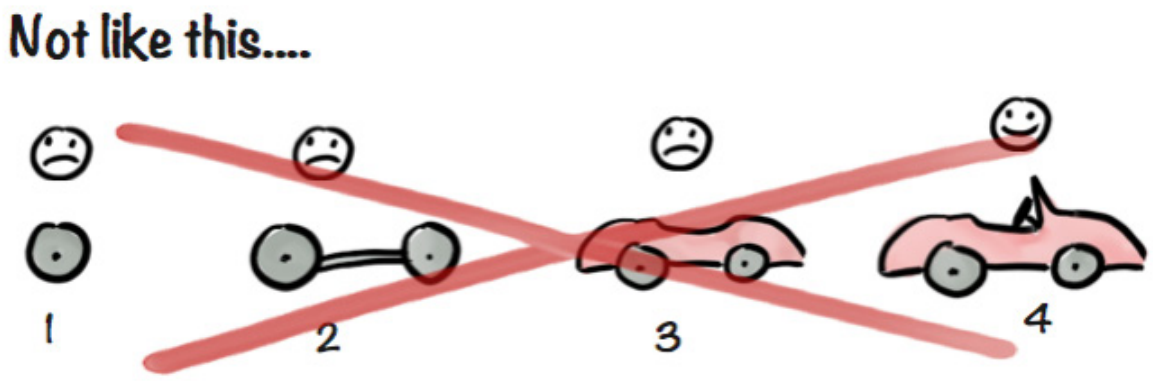
Ultimately, it leads to a lower cost, lower risk, higher quality product. We hope Spotify's strategy has helped you better understand the roles that Lean, Agile, and MVP thinking play towards streamlining your own product development. Don't be afraid to fail fast and quickly — as long as you keep testing, your most profitable idea might just come from what you learn.

CHAPTER EIGHT

4 REASONS MINIMUM VIABLE PRODUCTS FAIL

Avoid the pitfalls to learn more, provide maximum value, and scale

It is common belief that roughly 80-90% of products fail. Although this figure defies logic, the myth persists.



Source: Stop Overthinking... Just Stop

If over 80% of new products truly fail, does it make sense for anyone to take a professional career risk by committing their time and talents to building new products? From a portfolio perspective, what must the ROI of the remaining 10-20% of successful products be to offset the costs of a development program that fails at that rate? Operationally, how long would it take to realize these returns and what manager would realistically employ time, talent and money on these new product projects?

Luckily, in reality, the failure rate is 30-49% across all industries, roughly 39-42% in software & services and technology industries, respectively, according to nineteen peer-reviewed research studies between 1945 and 2004 (sources 1 & 2). Although the rate of failure is dramatically lower than what is rumored, the many risks of product failure cannot be understated.

Below, I talk about the many ways products can fail, how to tackle those risks systematically, and how UXPin has done so over the years.

SYSTEMATICALLY TACKLING 3 TYPES OF RISK

Ash Maurya, CEO of Cloudfire and author of *Running Lean - Helping Entrepreneurs Succeed*, offers a lot of advice about products, especially MVPs. He's covered topics such as how to build an MVP, delivering customer value, product launches and identifying the riskiest assumptions. On the last topic, he claims there are three general approaches product leaders can take:

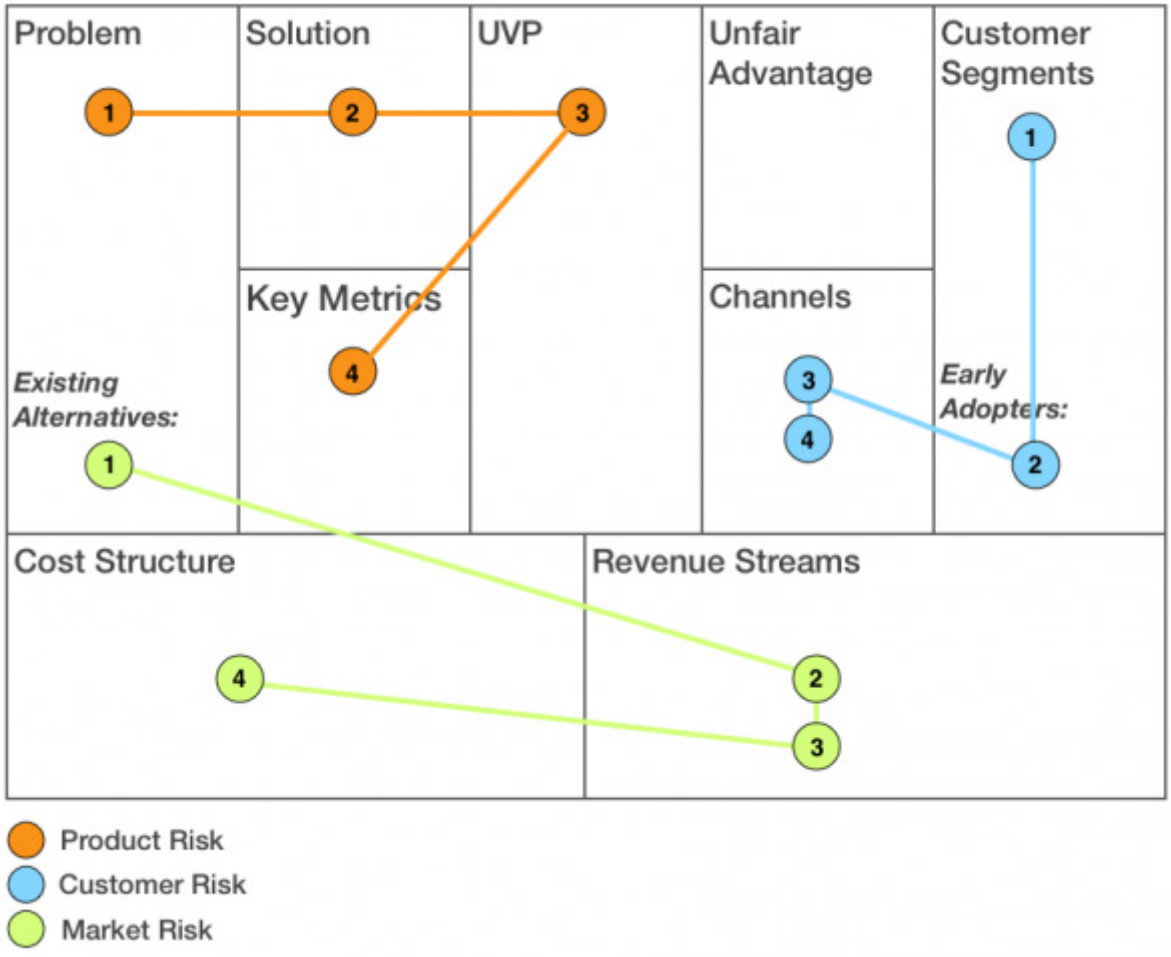
1. Use your intuition
2. Start with the 3 universal risks — I've detailed 4
3. Talk to domain experts

Ultimately, you alone have to own your business model and product roadmap so having a systematic approach to evaluating them throughout the business and product lifecycle is imperative. Since the first and last approaches are arguably more risky — especially at the earlier stages of a product — I'll only elaborate on the second approach using the Lean Canvas framework below for nailing your product or MVP.

In his post about 10x Product Launches, Maurya describes in detail how to address the 3 universal risks and I've summarized them below. This focused and systematic thinking about product risks was influenced by the management philosophy Theory of Constraints (TOC). Adopting the common idiom "a chain is no stronger than its weakest link," this management paradigm assumes that processes, organizations, etc. are vulnerable because the weakest person or part can always damage or break them or at least adversely affect the outcome. As a result, managers put this philosophy in practice by focusing on a very small number of constraints that could limit their business or product "system" and restructuring those systems around the constraints to achieve more goals.

I've summarized the steps below:

1. Identify the system's constraint(s)
2. Exploit the system's constraint(s)
3. Subordinate all other resources to the constraint
4. Elevate the system's constraint(s)
5. Rinse and repeat



Source: The 10x Product Launch

PRODUCT RISK – GETTING THE PRODUCT RIGHT

- First make sure you have a problem worth solving.
- Then define the smallest possible solution (MVP).
- Build and validate your MVP at small scale (demonstrate UVP).
- Then verify it at large scale.

CUSTOMER RISK – BUILDING A PATH TO CUSTOMERS

- First identify who has the pain.
- Then narrow down to early adopters who really want your product now.
- It's okay to start with outbound channels.
- But gradually build/develop scalable inbound channels – the earlier the better.

MARKET RISK – BUILDING A VIABLE BUSINESS

- Identify competition through existing alternatives and pick a price for your solution.
- Test pricing first by measuring what customers say (verbal commitments).
- Then by what they do.
- Optimize your cost structure to make the business model work.

HOW TEAMS FAIL TO ADDRESS THESE RISKS

An MVP can fail for myriad reasons. Some of these failings have been highlighted in the Forbes' article, 8 Reasons Startups With Good Ideas Fail, but we've got in

far greater depth below.

PRODUCT FAILURE

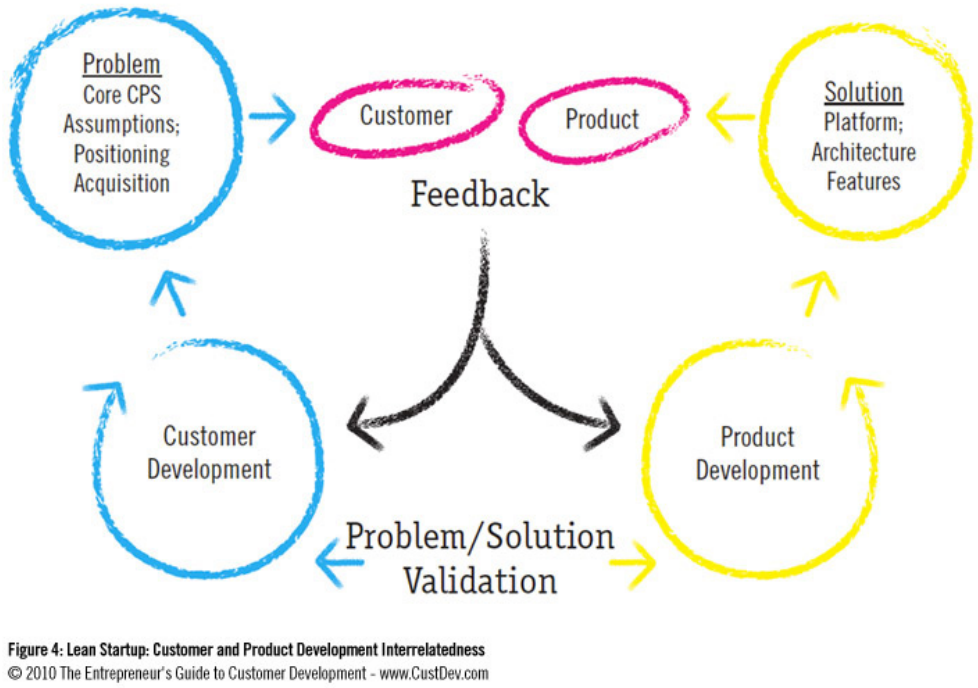


Figure 4: Lean Startup: Customer and Product Development Interrelatedness
© 2010 The Entrepreneur's Guide to Customer Development - www.CustDev.com

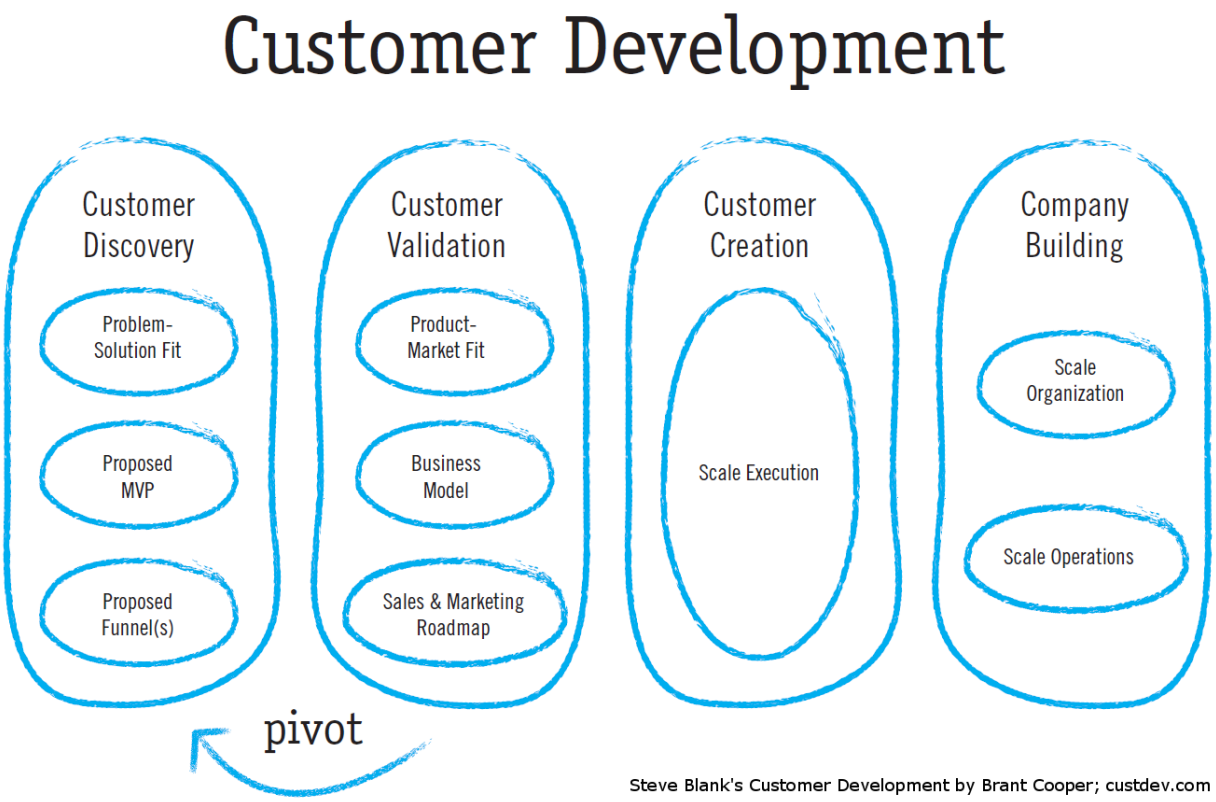
Source: LeanBlog

- **Solving the wrong problem** — Overlooking the purpose of the product before building, or simply focusing on the wrong purpose. You'll either end up with a product nobody cares about or one that doesn't give you the right answers.
- **Solving the worthless problem** — Worse than having the wrong product that can be tweaked (or pivoted) to solve the right problem is solving a problem correctly, but it's so minor in people's everyday life that it ultimately can't be sustained.
- **Poor customer communication** — You can't solve someone's problems cor-

rectly if you can't ask them the right questions.

- **Not translating customer problems into correct product requirements** — You understand the problem that needs to be solved but end up building the wrong solution.
- **Not iterating on solutions** — You understand the problem being solved and have a great idea for how to build a solution, but you don't consider alternatives that could be far better and would give you better answers to the questions you're trying to solve. Lack of brainstorming, customer engagement, sketching, and wireframing and/or rapidly prototyping could hurt you.
- **Overbuilding** — You figure out the right solution but add too many features that clutter the valuable functionality, leading to customer abandonment or confusion which will also impact the quality of your learnings because the user feedback and data is so scattered it doesn't make sense.
- **Launching late** — Again, you know the solution but then spend too much time building before launching your product. You either become disconnected from the customers you're trying to satisfy or run out of time, money and resources.
- **Lacking data** — You're not tracking or, worse, you're improperly tracking user behavior so you can't properly make informed decisions and ultimately (in) validate your hypotheses about the MVP. You just move onto the next product release with only one eye open.
- **Lacking scale** — If you don't get enough data points, your hypotheses about what you're building is harder to (in)validate. This is more important the bigger or more sophisticated your product becomes because it's harder to get signal from all of the noise for the growing number of questions you have about the product.

CUSTOMER FAILURE

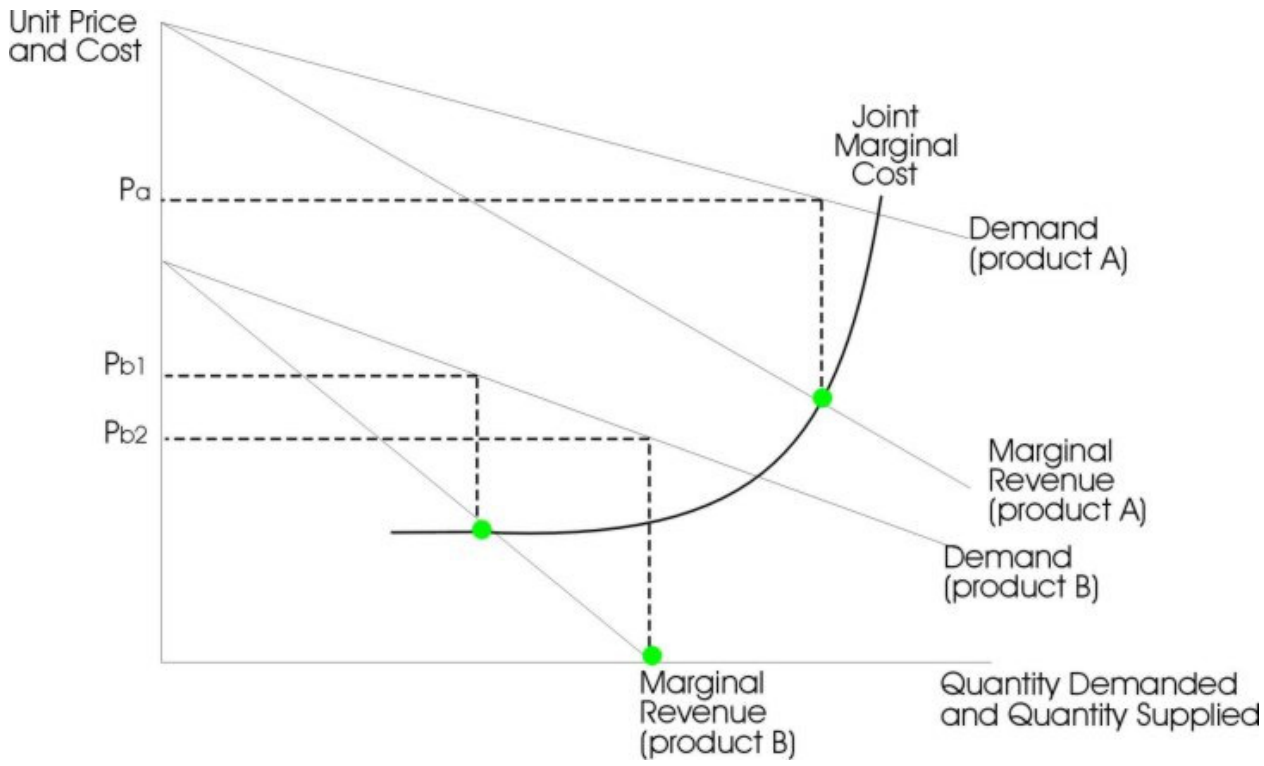


Source: "What is Customer Development?"

- **Solving one's own needs, not customers** — It's one thing if you built a product that solves your problems and it's another if the product ***only*** solves your problems. You need to socialize your product with others to find who actually needs your product. Then figure out where you can find more of them.
- **Building for too broad of an audience** — You can't be all things to all people. You end up being a "Jack of all trades, master of none." The results of this are twofold: You'll have a product that solves more people's problems to a lesser degree but you'll also have a harder time figuring out who to go after first. After all, it takes time, money and energy to get anyone to become your customer, and the task is even more daunting when you're trying to make the entire World your customer.

- **Not finding early adopters** — Focusing on the wrong early customers can be just as bad as not having focus. If your first customers are people who aren't in love with your product, then your product either sucks or you haven't found the early adopters yet. Think about it. You're solving a problem that led you to put every day of your life into building a product. You want people that would use your product every day of their life — or close to it.
- **No plan for marketing and user acquisition** — You have early adopters who love your product but no real way to get your product to market. Your great product will die a slow death from neglect.
- **Pushing to customers, not pulling them** — You can only scale so much by cold calling and emailing people. Even if your product is successful early on, you'll need to find better ways to get inbound interest.

MARKET FAILURE

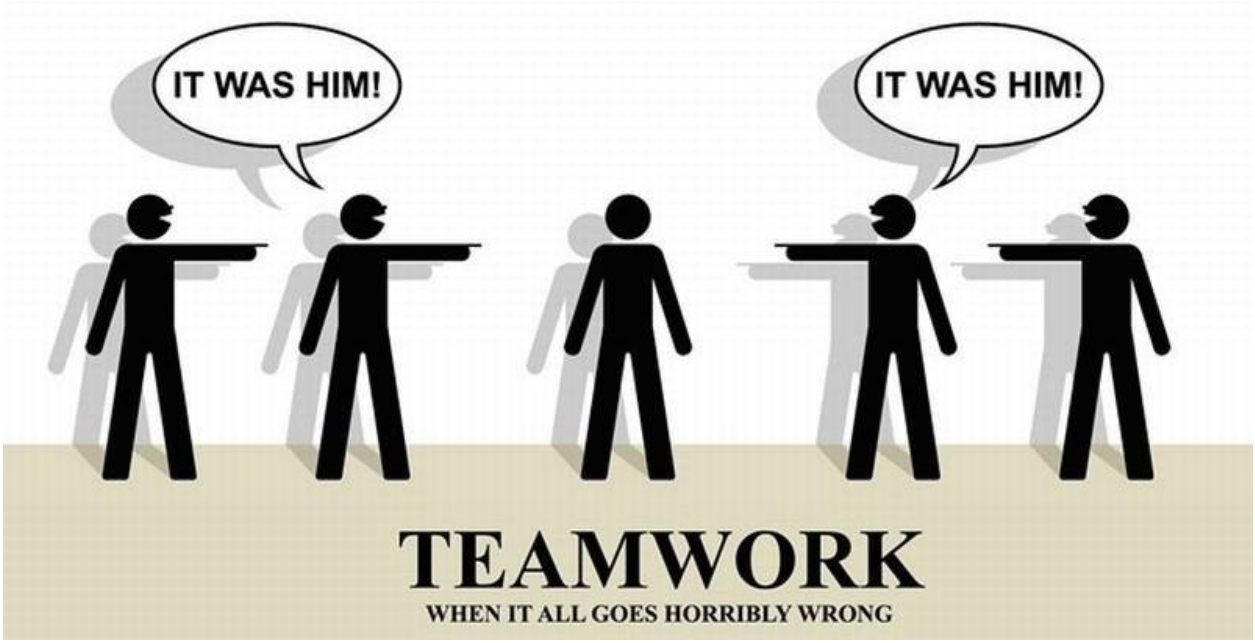


Source: Wikipedia

- **Not considering alternative solutions** — You understand the problem being solved and have a great idea for how to build a solution, but you do it in a vacuum. Consider that customers have the power of choice and that you have to compete for their business. If it's a big problem, you're not the only one trying to solve it.
- **Not charging** — It's hard to get a sense of what your product is worth to customers if you don't charge (or at least ask them verbally early on). It's also impossible to sustain your product if you don't make money. While ad-based businesses often start out free or freemium, they still have a path to revenue. You should too.
- **Not pricing correctly** — Charging improperly can be as problematic as not charging at all. You could have a false sense of what your product is worth and not catch this for a long time. And it's always changing based on the next best alternative, customer behaviors and many more factors. Don't just throw discounts at customers to get them in the door. Try to really understand what your customers will pay. This could also lead to failure for many reasons.
- **Not tying pricing to customer value** — Along the same lines, your product typically provides different value to different customers for different reasons based on specific functionality you've built. Your product doesn't offer a standard utility to everyone. Therefore, it's important to really understand how specific aspects of your product can be segmented to get the most revenue and deliver the greatest customer value.
- **Not adapting the business model to be profitable** — You may have to find a new customer or a new way of getting money from your existing customers even if the product is serving a really important need.
- **Not adapting the product to be profitable** — Some parts of your product may make it really expensive to operate, may cause a lot of customer service

maintenance, etc. Your product can impact your ability to sustain itself. It's not just the business that could be wrong.

TEAM FAILURE



Source: thezooom.com

While Maurya doesn't include this in his framework, the human factor in product failure is critical to consider. Forbes' article, 8 Reasons Startups With Good Ideas Fail, is a good starting point. I've laid out some of the main points of failure below:

- **Not taking action** — Ideas and direction are worthless if nobody can deliver on them. Don't get too far ahead of yourself with your ideas (and idealism). You'll spin your wheels and spend your resources without getting anything tangible done. Or someone else will come in and take your spoils because you were too busy brainstorming release 10 when you haven't finished v.1.
- **Giving up** — You just stop, call it quits, throw in the towel. You give up. Building an amazing product and business is hard. Don't feel bad if it happens. Just

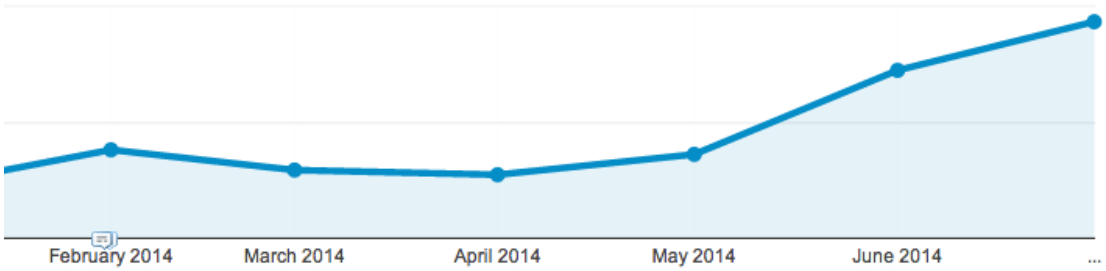
recognize that it happened and don't make excuses.

- **Lack of expertise** — Collecting data and talking with customers is great. But if you don't know how to interpret the feedback, you'll be blind. And the blind leading the blind may drive you off a cliff.
- **Disagreement about goals and/or direction** — If you don't have a clear objective for your MVP across the team, then your methodology and, therefore, conclusions may be compromised.

STAY FOCUSED OR YOUR MVP WILL SUX

Any of the above MVP failures (or a combination thereof) will produce a product that SUX — an offering with a "Sh***y User Experience."

At UXPin, we try to address all of these potential failing points. Like many companies, we've certainly been guilty of building features that weren't the most important to our customers or, ultimately, company growth. However, we've been aware of and addressed many of the potential failure points mentioned and the results have been incredible. While I can't give you exact numbers, you can at least see that we're driving significant changes in one of several key metrics over the past few months and are on track to blow that number away as well. We're clearly doing something right.



Source: UXPin

CHAPTER EIGHT

10 MASSIVELY SUCCESSFUL MINIMUM VIABLE PRODUCTS

Hot products that have mastered the MVP since the beginning

Today, lean startups and tech titans alike are increasingly using the minimum viable product (MVP) as a starting point for building successful software products. A successful minimum viable product helps you start the learning process as soon as possible, and not just to answer the technical questions of "how" but also to eliminate the business uncertainty of "why."

Now let's take a look at some of the companies that got their MVP right and what they did to go on and launch some of the hottest products in the market today.

1. DROPBOX

In his book The Lean Startup, **Eric Ries, cofounder/CTO of IMVU** talks about how Dropbox tackled the question of market viability by demonstrating their product in a video.

To answer the question of whether customers would want to use and pay for their file-sync solution and to justify the market to investors, Houston and his team had to "get out of the building" and put their proposed user experience in front of actual users to get feedback. Instead of digging into servers and building a high-availability, low-latency, always-on network even before they had any clue people would use it, the team decided to try something else.

Source: Dropbox

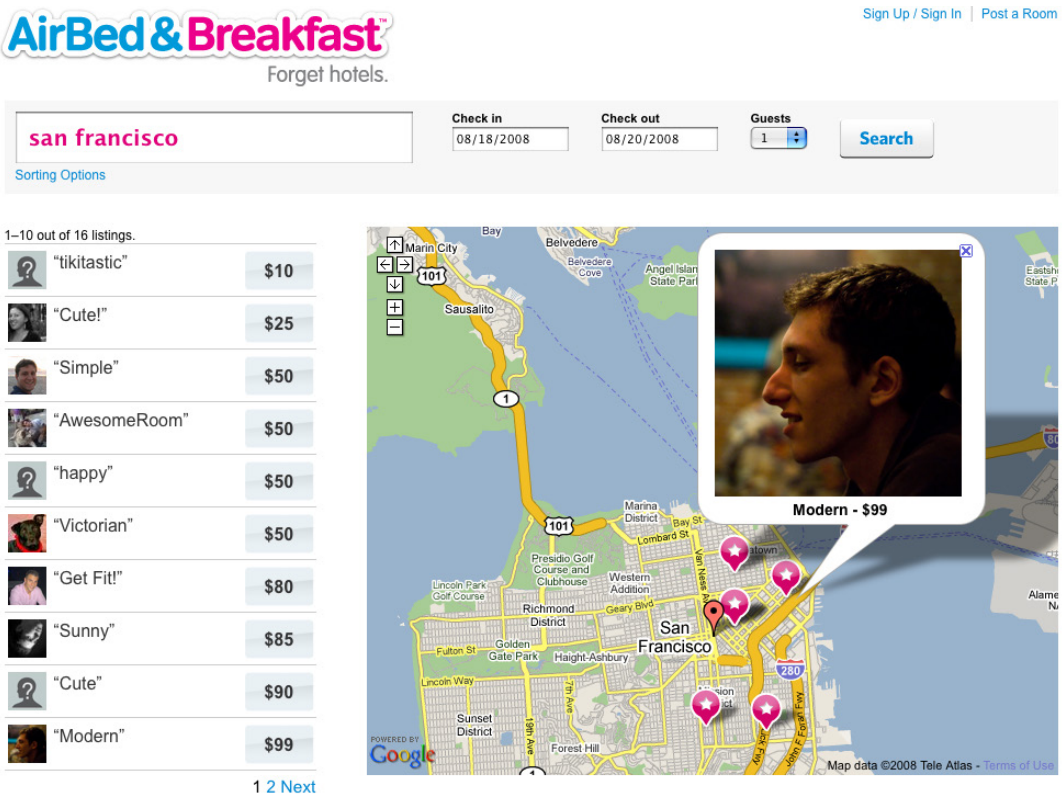
They made an **explainer video** and started sharing it with their network to see how people would react. The 3-minute video demonstrated Dropbox's intended functionality and resulted in signups increasing from 5,000 people to 75,000 overnight — all of this in absence of a real product.

Dropbox's explainer video served as a brilliant validation of the market before the founders ever had to invest in the infrastructure and development needed for a high-tech product like theirs to reach a functional level in the real world. It walked potential customers through what the product is and clearly demonstrated how it would help them, eventually leading to why they would want to pay you for it.

When it comes to product development, it's easier said than done but when you're building an MVPs, that's not necessarily a bad thing.

2. AIRBNB

In 2007, Brian Chesky and Joe Gebbia wanted to start a business, but also couldn't afford the rent of their San Francisco apartment. There was a design conference coming to town, and they decided to open up their loft as cheap accommodation for conference attendees who had lucked out on the hotels nearby. They took pictures of their apartment, put it up on a simple website, and soon they had 3 paying guests for the duration of the conference: a woman from Boston, a father from Utah, and another man originally from India.



Source: [AirBed And Breakfast Takes Pad Crashing To A Whole New Level]

The up-close interaction gave Chesky and Gebbia valuable insight into what potential customers would want. This **concierge MVP** helped validate the market and prove people would be willing to buy the experience. With their initial assumptions answered that not just recent college grads would be willing to pay to stay in someone else's home rather than a hotel, they started Airbnb (then called AirBedAndBreakfast).

3. GROUPON

Andrew Mason started with a website called The Point, a platform to bring people together to accomplish things they couldn't do alone, like fundraising or boycotting a retailer. But the site wasn't gaining much momentum, so they decided to try out something else.

Source: The Point

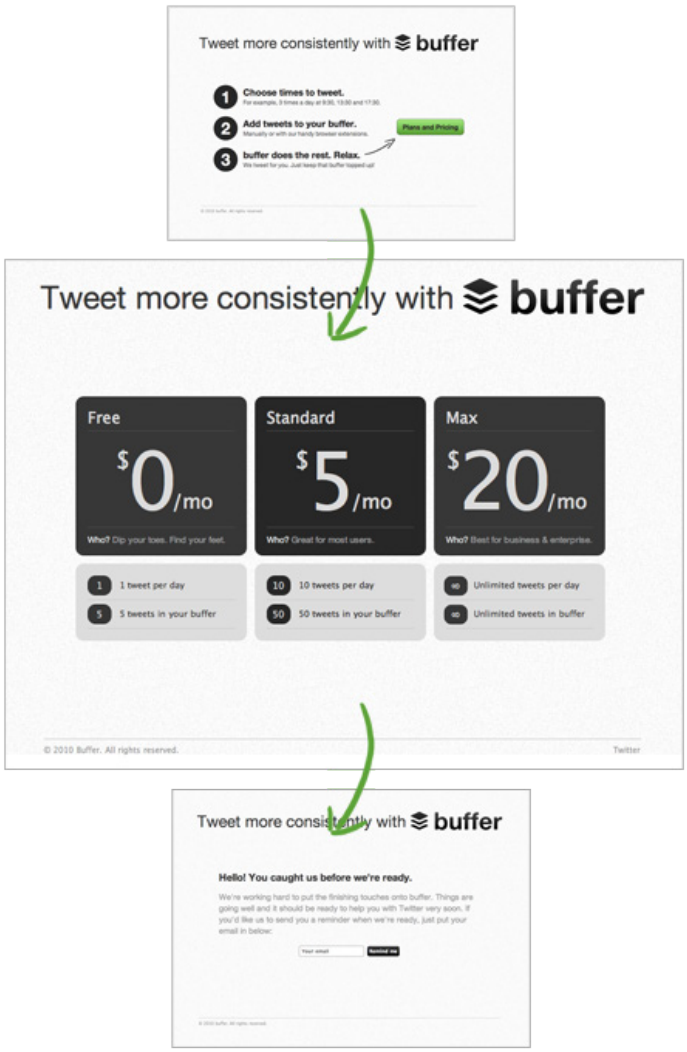
Using the same domain, they set up **a customized WordPress blog** called The Daily Groupon and began posting deals each day manually. When someone signed up for a particular deal, the team would generate a PDF document and

email it using Apple Mail. This simple website they "cobbled together" showed the team this was a market worth looking at with just a **manual-first ("Wizard of Oz") MVP** that helped them pivot their offering from what they had been doing previously.

They didn't invest any time into developing a coupon system and designing a new website. Instead, they took what resources they had and made a **piecemeal MVP** out of them to test the hypothesis of whether people would be interested in what they were offering. Starting from a customized WordPress website and manually emailing PDF documents to a mailing list isn't exactly what you'd call scaleable, but Groupon's MVP was successful in answering that question for them.

4. BUFFER

Buffer is a simple app that lets you schedule your posts across your social network, essentially letting you space out your updates so that you don't flood your friends' newsfeeds at one point in the day with interesting stuff you want to share. When starting out, Joel Gascoigne, Buffer's founder, didn't want to get stuck building a product no one wanted to use. So he began with a simple test.



Source:: [Idea to Paying Customers in 7 Weeks- How We Did It]

[Buffer's first minimum viable product] was just a simple **landing page**. It explained what Buffer was and how it would work, encouraged people to sign up and offered a plans and pricing button for people to click on if they were interested. When they did, however, they were shown a short message explaining they weren't quite ready yet and that people should sign up for updates. Joel used the email addresses received from the signup form to start conversations with the potential users of the app, gaining valuable feedback and insight into what they would want.

Next, they tested the hypothesis that people would want to pay for this by add-

ing the prices table in between the landing page and the signup form. When someone clicked on the pricing plans button, they were shown the plans to see whether they would be interested in paying for something like Buffer. This showed Joel how many of the visitors to the site could potentially become paying customers. This zero-risk MVP helped Buffer identify the market and shape their product features in the coming development as well.

5. ZAPPOS

Today, we know that people are comfortable with buying shoes online when Zappos had annual sales of more than $1 billion and was acquired by Amazon for $1.2 billion in 2009. But in 1999, when co-founder Nick Swinmurn wanted to build an online retail store that stocked a great selection of shoes, the assumption that other people would use it needed to be tested.

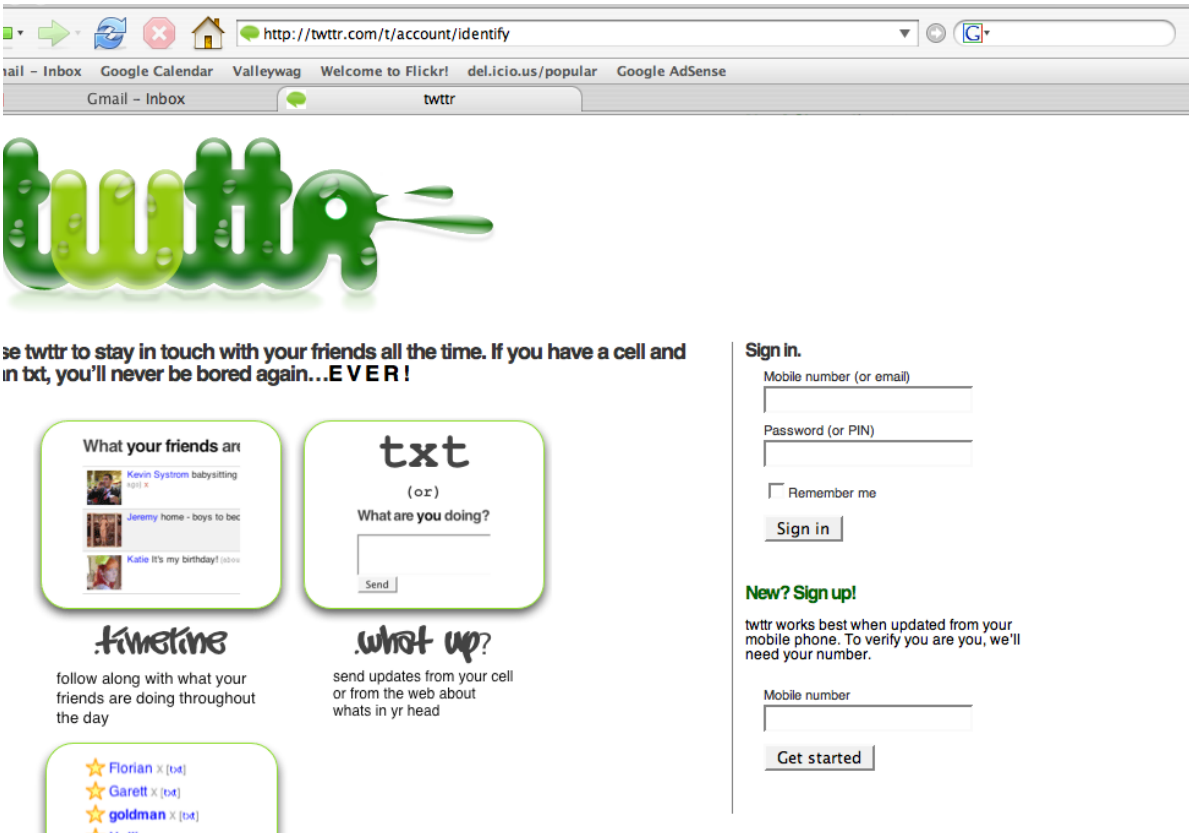
Source: Zappos

Swinmurn began by putting up photos of shoes from local shoe stores on a website to gauge demand for an online store. When someone ordered the shoe online, he would return to the store and buy it. Instead of first investing in infrastructure and inventory, this gave Zappos a chance to answer the question of whether their product would be accepted by the market. This **Wizard of Oz**, man-behind-the-curtain technique is used to test market hypotheses. It comes from the idea of putting on the impression of full functionality, essentially faking it until you make it. Customers believe they are experiencing the actual product, but in reality the work behind the scenes is being done manually. In the case of Zappos, by Swinmurn.

The Wizard of Oz approach is great because it also allows for greater interaction with customers at this crucial stage when you're designing the product. The expedited learning, albeit at a small-scale, provides opportunities to test many assumptions you might be making about the product or the marketplace.

6. TWITTER

Twitter traces its origins to the podcasting platform Odeo. When Odeo found the ground underneath slipping away after Apple stepped into the podcasting game with iTunes, the company started running hackathons to come up with ideas of where to head next. One result of this was an idea for sharing updates with a group of people via text messages, codenamed "twttr".



Source: [Famous First Landing Pages]

The first **prototype** was used as an internal service for Odeo employees, who

eventually became so obsessed with it that they began racking up “monthly SMS bills totaling hundreds of dollars”. This gave the team the push they needed to release Twitter to the public, but it was only at the SXSW festival in 2007 that their user base exploded when they showed off members’ tweets about the event on TV screens across the venue.

7. ZYNGA

Zynga is a game studio that builds social games, popularized by the likes of Farmville, which surpassed $1 billion in revenue from in-game purchases in 2013. For their game development process, the company follows a mix of **landing pages** and **Adwords MVP tests** to gauge interest in a planned game or particular aspect of the game.

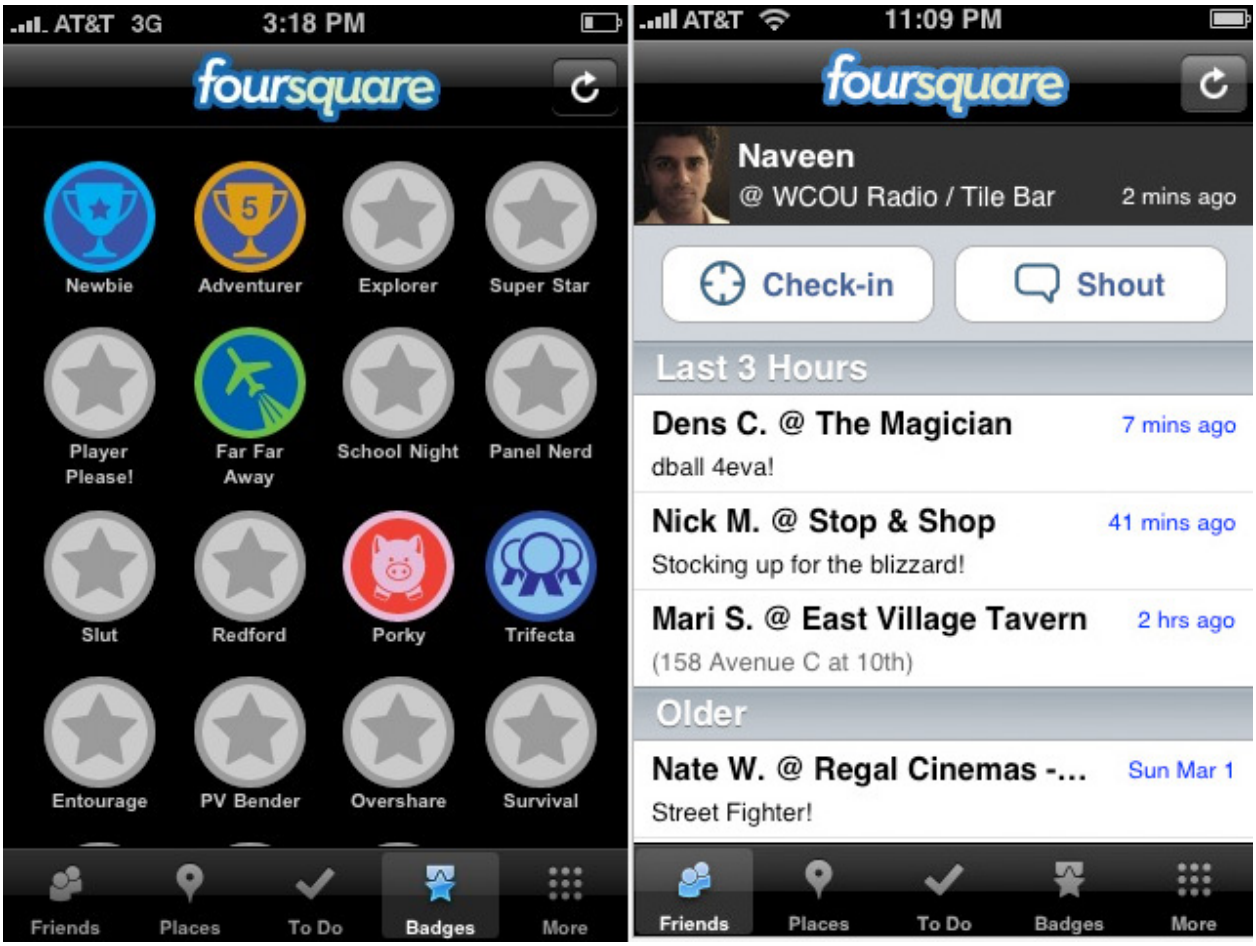
Source: [Zynga Rewardville](#)

By running short ads in existing games and online that pitch potential game ideas and features, the company is able to gather data about which direction to steer development towards, preventing them from wasting any resources on building a

game that people don't end up playing.

8. FOURSQUARE

Foursquare is a location-based social network that lets users check-in with their location to share with friends and family. After their first such network, SMS-based Dodgeball, was acquired by Google, founders Dennis Crowley and Naveen Selvadurai set out to work on a mobile app based network they called Foursquare.



Source: [Foursquare scores despite its flaws]

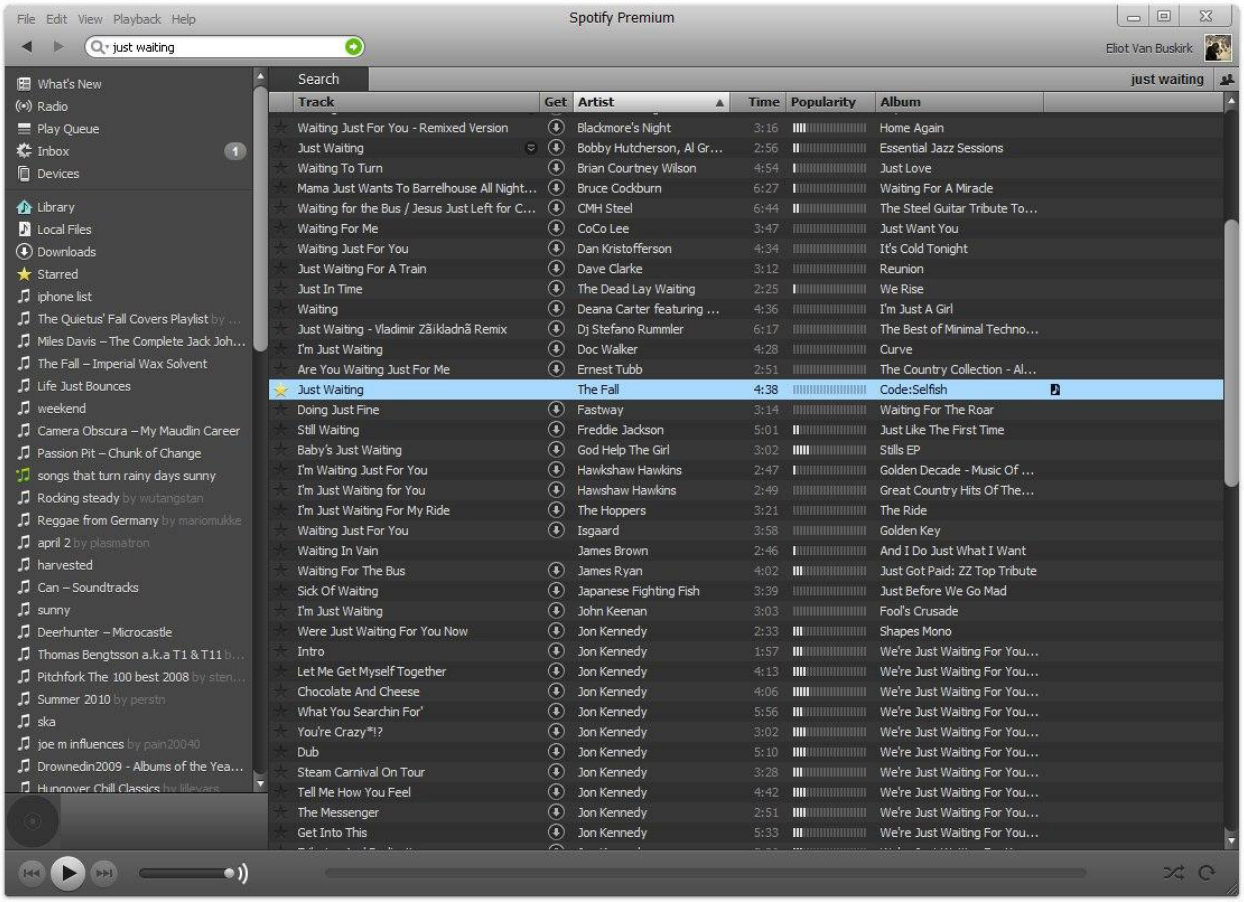
Backed by VC funding, they didn't let development time slow them down, however, and launched with a **single-featured MVP** that didn't bog them down in

design and unnecessary features. They began with check-ins and the gamification rewards and focused on improving that side of the user experience, using feedback to mold their product.

Once they were comfortable with the basic functionality, they began adding extra features like Recommendations and City Guides years later. Additionally, they used **existing services** to help manage the service, for example using Google Docs to gather feedback and requests from users, which goes to show that you don't need to do everything yourself when you start in order to build a scalable business.

9. SPOTIFY

According to Henrik Kniberg, Agile and Lean Startup consultant and author, Spotify uses a 4-stage iterative product cycle (Think It, Build It, Ship It, Tweak It). When they launched in 2009 with a **landing page,** they focused on the **single feature** that mattered most: music streaming experience. With the desktop apps, they were able to test the market in the limited beta run, giving them time to build momentum to tackle the music industry licensing concerns that were sure to come as they planned on expanding to the US.



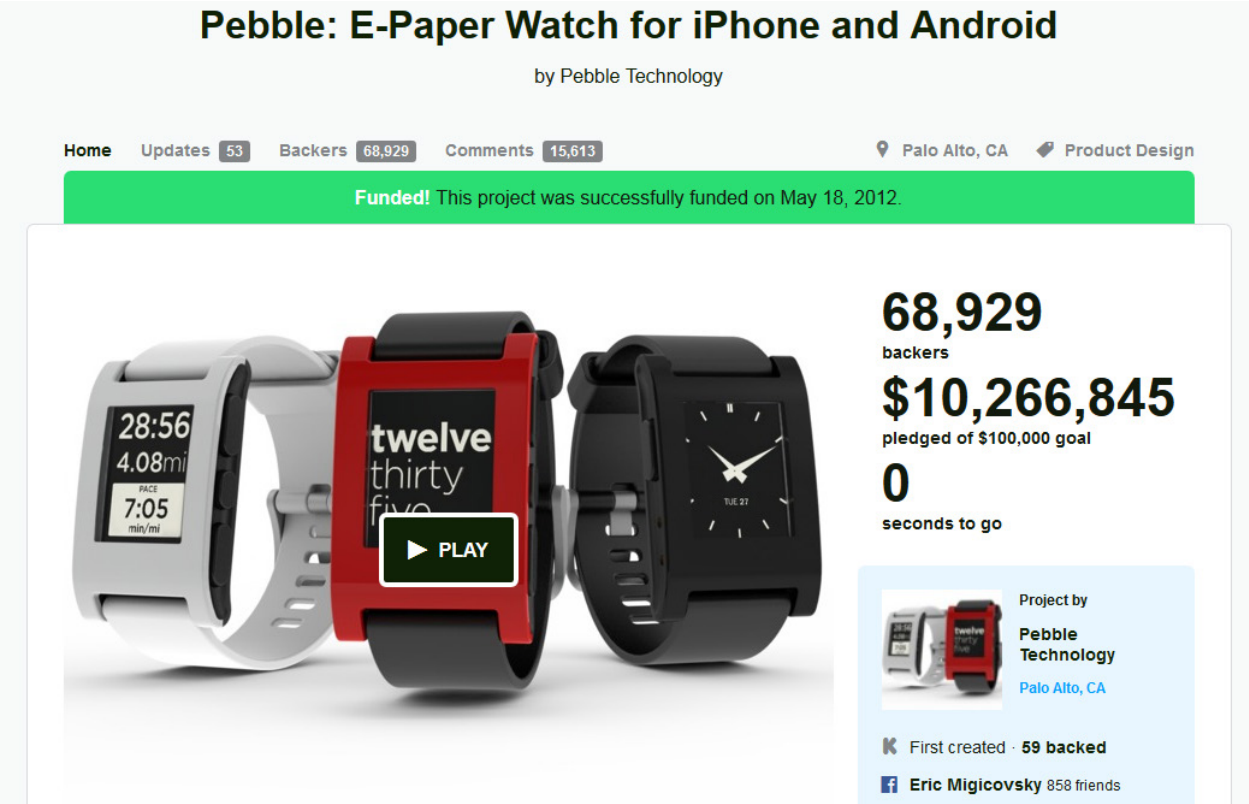
Source: [Spotify set to take America by storm]

They follow the same Agile process today to help scale. Each one of Spotify's four stages are Lean since small teams are always working smartly to test assumptions. The "Think It" stage tests the merit of conceptual MVPs while the "Build It" stage releases a physical MVP only after it's been tested for quality. The "Ship It" and "Tweak It" phases ensure long-term quality and customer alignment by releasing the MVP gradually, learning from feedback, and iterating tirelessly.

10. PEBBLE

Pebble is an e-paper smartwatch, arguably the one that brought the "wearables" market to the mainstream today. After investor funding dried up, founder Eric

Migicovsky turned to crowdfunding site Kickstarter for **fundraising** and ended up becoming the most successful project there, raising more than $10 million from interested customers looking to support development.



Source:Pebble, E-Paper Watch for iPhone and Android

Migicovsky recorded an **explainer video** to demonstrate the prototype and asked interested customers to contribute. They reached the original goal of $100,000 in 2 hours and by the end of the week had raised $600,000. When the funding round on Kickstarter ended, more than 60,000 people had pledged $10.2 million to the project and Pebble went on to develop the watch for consumers. As of March 20, 2014, Pebble has sold over 400,000 units.

TESTING THE RISKIEST ASSUMPTIONS

In his book, The Lean Startup, **Eric Ries** writes about how to choose what to test

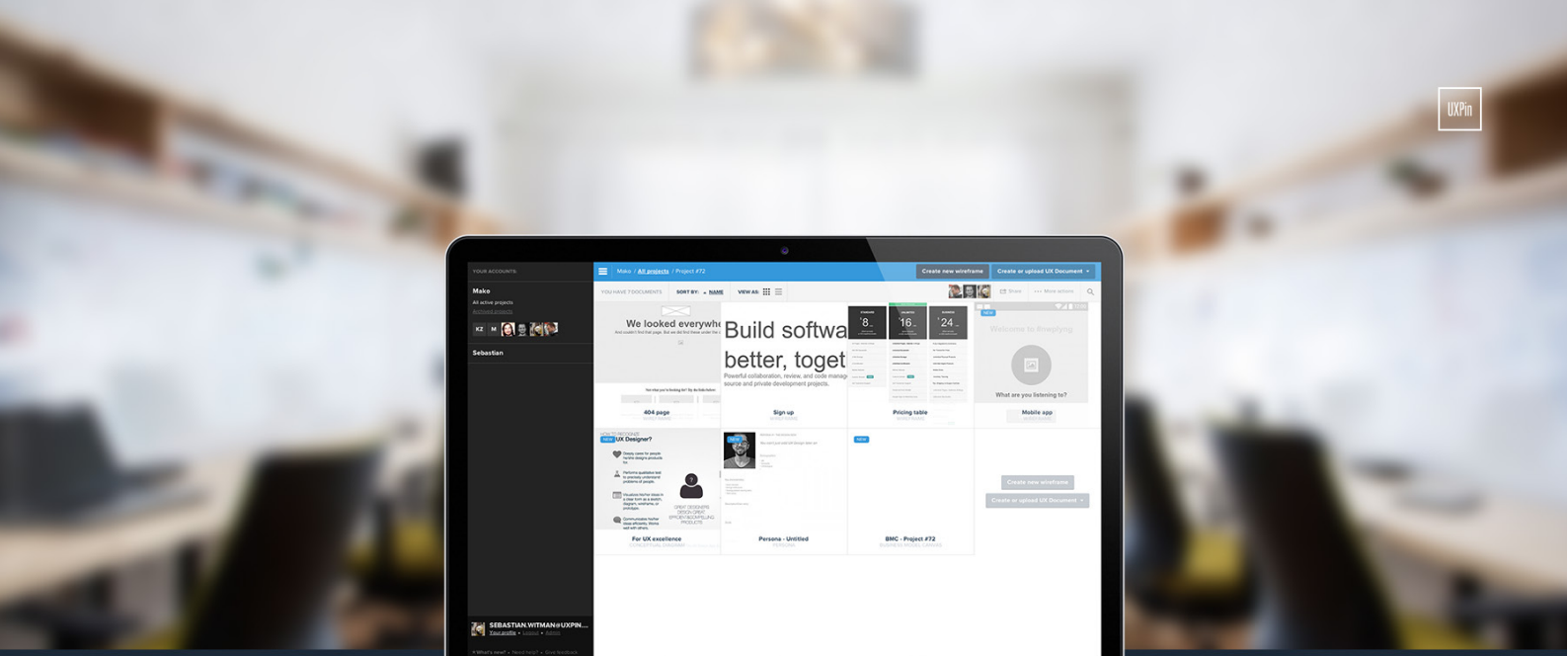
when designing an MVP:

"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 most startups, the riskiest assumption is the existence of the market. For Dropbox, that assumption was that people wanted to use a file synchronization service. For Zappos, it was that people would buy shoes online. For Airbnb, it was that they would be willing to live at a stranger's house as opposed to in a hotel. And in each of these cases, they designed their MVP to answer the important questions on which their businesses rested.

As you've seen, the minimum viable product is more a way of thinking than something that's just released to the market. The principles are the same whether you're a Fortune 500 company headed for your next breakthrough or a one-man shop on the verge of putting an "Aha!" moment into action. Don't dive into the deep end with all your resources strapped around your waist. Research thoroughly, build features moderately, and keep the team focused by working in small batches. Above all else, test, tweak, and test some more.

I've done enough talking. And you have ideas to validate (or invalidate). Now go forward and use what you've learned to test and refine your vision.



- ✔ Complete prototyping framework for web, mobile and wearables
- ✔ Collaboration and feedback for any team size
- ✔ Lo-fi to hi-fi design in a single tool

UXPin

www.uxpin.com