A Path to the Minimum Viable Product

I first met Shawn Carolan and his wife Jennifer at the turn of the century at 11,000 feet. I was hiking with my kids between the Yosemite High Sierra camps. Having just retired from a career as an entrepreneur I had started thinking about why startups were different from large companies. The ideas were bouncing around my head so hard that I shared them with these strangers around a campfire, drawing out the four steps with a stick in the dirt. Shawn immediately said the name I had given the four steps was confusing – I had called it market development – he suggested that I call it Customer Development – and the name stuck. What I didn't realize was that both were graduate students at Stanford and later both would become great VCs – Shawn at Menlo Ventures and Jennifer at Reach Capital. (And Jennifer is now my co-instructor in the Stanford Lean LaunchPad class.)

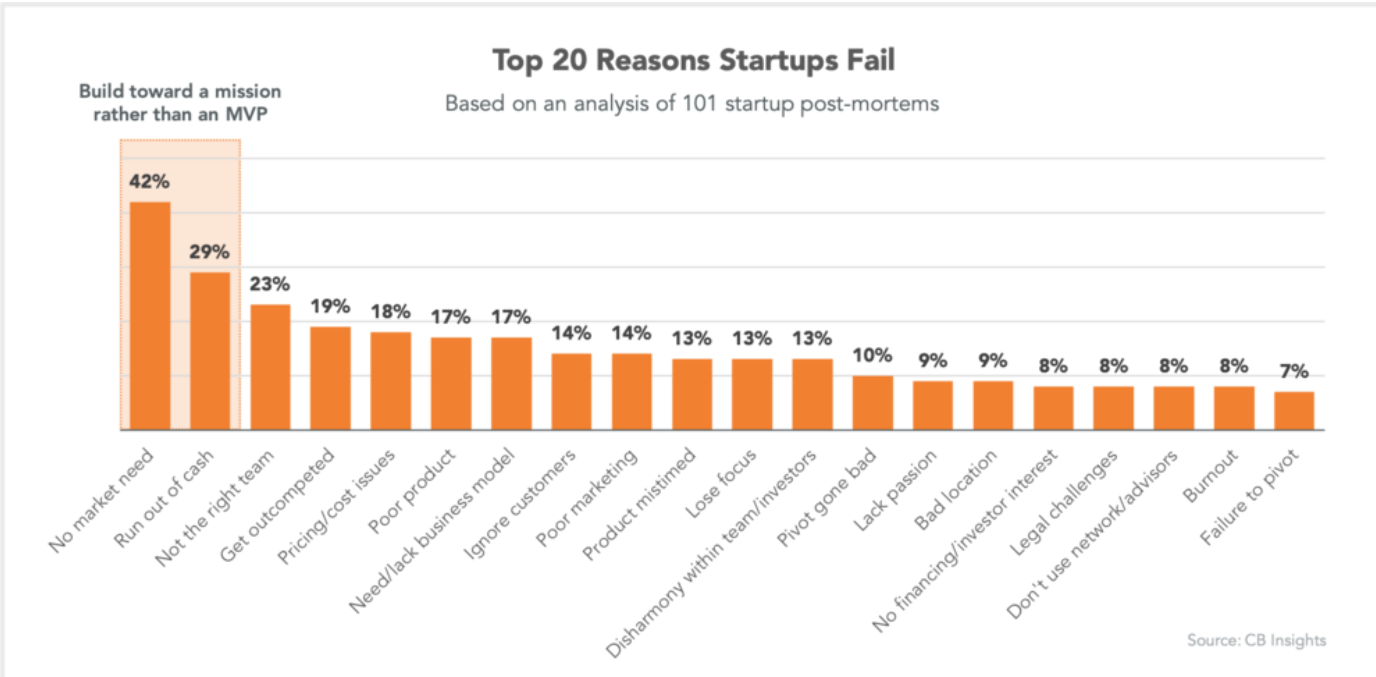
The MVP Tree

Over the last two decades Shawn has seen hundreds of startups use the Lean Methodology. Many of them get hung up on understanding how to select the right minimal viable product. He came up with the concept to simply the search for product/market fit by using an MVP Tree.

Shawns guest blog post describing describing the MVP Tree is below.

(Note that if you're familiar with the business model canvas, Steps 1-4 below are equivalent to a visual map of the choices a founder makes as they develop a business model canvas. Step 5 and 6 leads you to selecting the right MVPs.)

It’s commonly believed that the top two reasons startups fail is because “there’s no market need” and “they ran out of cash.” These reasons are mental gymnastics to avoid a plain truth**:** startups fail when they don’t build a simple solution to a problem many people have.



Many startups fall into the trap of building toward a “mission” rather than a minimum viable product (MVP).

Your mission is your baby. It’s the North Star that got your people on board and inspires them daily. However, solely focusing on your mission is the same as being unfocused.

Bridging the gap between a big, ambitious dream and the reality of what you can accomplish with limited resources isn’t fun because it requires saying no (for now) to a lot of things that have you excited. Paradoxically, resource limitations are the secret to success; they teach lessons that experienced entrepreneurs have harnessed. Constraints force you to pause—or even permanently shelve—certain aspects of your mission in favor of proving that you can deliver one specific thing that really matters to customers.

During the crucial mission-to-MVP planning phase, the objective of a startup is to solve one job for one customer group such that customers consistently use your minimally viable product for an important part of their work or personal lives. In other words, you prove retention. It's really all that matters at the earliest stage.

The tool for doing this efficiently and effectively? We call it an MVP tree.

Solution: Build an MVP Tree
Company missions tend to fall to the extremes: either the mission isn't ambitious enough or it's too ambitious to build with your current resources.

It bears repeating: an early-stage startup must focus on *making one customer group excited by a mission-aligned product***.** Doing so is usually a long way away from realizing your full mission statement, and that's okay.

An MVP tree is a way of methodically breaking your mission into smaller components and formulating *MVP candidates* that may get your company sustainable and scalable. Using the tree structure outlined below increases the chance that your first step forward (the MVP) will be successful with a small team, while taking you in the right direction to achieve your company's big mission. An MVP tree has three main components—customer archetypes, jobs to be done, and execution—and we'll walk you through them step by step with a case study of Roku, a former portfolio company we've been fortunate to see execute from a $20 million first VC round to what is now a $50 billion public company.

(To get started, we draw our tree out using [Whimsical](), a product that I'm a big fan of.)

Step 1: Define Your Big Mission in a Simple Statement

In the day-to-day fray of prioritizing features, considering customer input, and handling people issues, it's easy for a team to lose their bearings. The world is full of great products and it's essential to be crystal clear on *your* reason for being; to avoid wandering in circles, you need a mission statement. A few examples:

- **Roku:** "To be the TV streaming platform that connects the entire TV ecosystem around the world"
- **Uber:** "To bring transportation—for everyone, everywhere"
- **Chime:** "We believe everyone deserves financial peace of mind"

Your mission statement needs to stand for something *specific and impactful*. It may change as you build and learn from your customers, but aim for it to conjure up an image of a better future with your product at the center.

Your mission statement is the center of your MVP Tree.

A motivating mission statement inspires action. It may be tempting to jump right into coding, but slow down—remember, the goal of this exercise is to determine *how little you can do* to *acquire and retain customers***.** With both growth and retention, you earn the right to build more.

Step 2: Customer Archetypes

A *customer archetype* is a category of similar people with similar needs (e.g., segmented by age, gender, profession, personality type, etc.). While it may be tempting to want to build a product for everyone, everywhere (7 billion TAM, right?!), doing so will distract you from building exactly what one group needs. Each new segment you attempt to serve can *increase scope*, adding to your workload and demanding more of your limited resources.

The intent of this branch of the tree is to identify one group of potential customers who would be highly motivated to fix their pain as part of your MVP. Draw out a few customer archetypes that might be a fit for your startup, (we'll discuss later how to prioritize the one group to tackle first.)

Step 3: Jobs to Be Done

Clayton Christensen's [Jobs to Be Done framework] proposes that focusing solely on customer data leads founders on a wild goose chase. Founders should understand *what the customer hopes to accomplish*, or what their *job to be done* is**.**

Break down your mission statement into different "jobs"; this itemization will likely narrow your product scope considerably, while still allowing you to create something of significant value.

When consumers have a job to be done, they'll look around at their choices and select the best tool for the job. Your goal is to build the best tool for meaningful, reoccurring jobs that people face. Map out the jobs that you believe exist for the customer archetypes identified in step 2.

The more common jobs may apply to several of the archetypes, while the more esoteric jobs may apply to just one. There will be plenty of overlap, which is why jobs have their own branch of the MVP tree rather than attaching directly to the archetype branches.

Step 4: Execution Branches

*Execution branches will vary based on the company, but think of them as the components of what gets built and where it gets sold***.** These branches of the MVP tree comprise the components of market expansion that urge founders to explore the tactical side of getting a solid product in front of the right people. This first product launch can either be a costly mistake or an ingenious first step that shows traction with early customers and gives your team and investors confidence. Map these out now as part of the tree and reduce the odds of a headache down the road.

In the Roku case study we've chosen three execution branches: (1) delivery platforms, (2) sales channels, and (3) chip platforms.

Delivery platforms

Delivery platforms are the vehicles through which customers come to interact with your product. For software products, they would be the operating systems with big market shares: iOS, Android, web, Mac, and PC. While supporting each additional platform expands your addressable market or breadth of customer touchpoints, it can dramatically increase scope. Developing for multiple platforms at once spreads already-thin resources, which ultimately harms the creation of the best product possible for a specific customer segment.

The fewer *platforms you choose to support, the smaller the scope. Pick one delivery platform to save resources and prove your value***.** Investors will recognize that a successful app on iOS will also work on Android with more capital. Focus your precious time on making one platform sing.

Clubhouse, a recent startup darling, has quietly grown to over 2M users exclusively on Apple. Some early users might be frustrated because they can't invite their Android peers, but the strategy to focus on iOS helped Team Clubhouse minimize their initial scope and meticulously learn from early users without the distraction that may come from opening the floodgates.

For hardware companies, delivery platforms are essentially the potential SKUs you might consider shipping. Steve Jobs once said that if you are really serious about software, you should build your own hardware. You can think of these hardware form factors as software delivery vehicles. Most people only know Roku for their TV devices, but Roku initially shipped an audio device for Internet radio stations and a PhotoBridge product to get digital photo libraries to the TV. Even when moving to video, consumers had the choice between a stand-alone box or a smaller plug-in stick. Today, Roku has become an operating system embedded in other brands' TVs.

Sales channels

Sales channels are the paths through which your product lands in customers' hands. For software products, the channels are typically through the mobile app stores or directly over the Internet. For hardware products, it's D2C e-commerce, online retailers, or physical retail.

Some sales channels may behave similarly; more often than not, they each pose unique challenges. Every *additional sales channel costs resources and increases scope***.** Pick one channel, prove traction, then experiment with the next.

Chip platforms

Unique to hardware companies is the choice of which chip to use, and it's a big one. The choice of semiconductor has profound implications on system requirements like how much memory is needed, how much power is required—and ultimately, the end system cost. Owners of the new Mac M1 laptops are taking advantage of a decade of Apple's mobile chip development, which is finally robust enough to run the MacOS. The Roku OS has come to run on several different chipsets over time, but in the beginning they had to choose one.

Other branches

There are several other execution branches that may be relevant to your business. If you are an office productivity tool, the data ecosystem you pick is a big one: Google or Microsoft? It's lazy thinking (and expensive engineering) to try to build for both at the same time. Pick one, show success, then raise more money to address the other half of the market. Again, the fewer of these branches (ecosystems, etc.) you choose to support, the smaller the scope. If your startup doesn't need to access customer data, the choice of consumer-grade vs. enterprise-grade is a big one that adds scope. The theme remains the same: be selective and pick the branch you have the best shot at success with before adding more.

Step 5: Scope Out Your Candidate MVPs

If your map looks like Roku's, you're probably now staring at something that has grown to become quite complex, with many "leaves" at the ends of your branches. A single leaf chosen on each branch is what makes up an MVP. The reasonable permutations are your *candidate MVPs*.

Remember, an MVP is **the** *minimally scoped product that gets some job done for your chosen customer archetype***.**

Step 6: Evaluate Your Candidate MVPs

How do you know which MVP to build first? Choose wisely: This will be the next several months of your life.
A successful MVP satisfies three criteria:

1. **It addresses a meaningful job to be done.** A customer spending their own time or money to do a job chooses your solution as the best option for them. Pick a meaningful job—the more frequently occurring the better—and offer a significant advantage (better, faster, or cheaper).

2. **It has a growth engine.** Build or price growth into your product. There are two viable growth engines for tech companies:

    1. **Viral, or "inherently viral" growth:** Customers either intentionally or unintentionally recruit new customers by using your product. Social platforms use intentional virality; this occurs when users get a more fulfilling experience as more of their friends join the same platform. Unintentional virality happens when customers inadvertently introduce others to the product experience, similar to how shared bike or scooter riders serve as mobile billboards for the experience.

    2. **Economic paid acquisition:** Contribution margins from customers are recycled into advertising, marketing, and other PR activities that successfully drive additional customers. Note well, though: this engine is "economic" because it must fuel itself. It's easy to simply buy customers, but only real value makes them stay. Your product must collect far more value over the lifetime of the customer relationship than the cost of acquiring that customer in the first place.

3. **It has a rapid time-to-value:** How long must customers wait for the "aha" moment? With my first Uber ride in late 2011, it took about two minutes for that moment to arrive: I installed the app, entered my credit card, ordered a car, and it was waiting for me by the time I walked down one flight of stairs. Aha! I knew I'd never wait for another taxi in San Francisco again. The faster and more simply your product can prove its worth, the higher rate of conversion from tire-kickers to retained customers. For software startups, ask yourselves this question: What's the **one screen that will make your customers get it?**

Step 7: Pick, Beta, Ship

Now's the time to let the rubber meet the road and get a minimal product into a customer's hands. Can it do the job better than their prior solution? Keep iterating until you're getting that one specific job done.

Roku's first two MVPs weren't a success (sound bridge + photo bridge). It was however, through the process of mapping out MVPs candidates, testing, and learning that brought intense clarity to and laid out the infrastructure for what would ultimately work — a delivery platform for Netflix. Even if the first MVP isn't a hit, you'll be building the muscle needed for the company such as making key hires.

Don't pivot to a different job unless you've learned something new that causes you to reconsider your initial hypothesis. This may take a lot of time, but that's perfectly fine. Stay focused on solving that job until you prove your hypothesis right or wrong. The world is littered with failed companies that never got a product right. Where there's a job to be done, you can build a solution with enough time, talent, and focus.

Roku's Winning MVP:

Step 8: Double-Down

When you finally find *your product getting a recurring job done better than any other tool*, stick with it. Don't take your eye off the prize and move onto new things too quickly.

Earn the right to increase scope and move on to other jobs, platforms, and customer archetypes after solidifying your position among this first set of customers—and creating a sustainable growth engine.

Final Thoughts

Founders become infatuated with a bold and ambitious mission—as they should. However, what separates a startup that actually brings its mission to life from one that doesn't is the ability to shed the rose-colored glasses and solve for a small job to be done.

A proper MVP framework, such as our MVP Tree presented here, is a critical first step in fulfilling your mission, even though it might seem like you are selling it short. Be patient. It won't be easy realizing your mission, and it shouldn't be**.** If your mission were easy, it would already be done by someone else!

Want to see an MVP Tree for another startup? Tweet @Shawnvc to nominate a company!

Filed under: Customer Development |