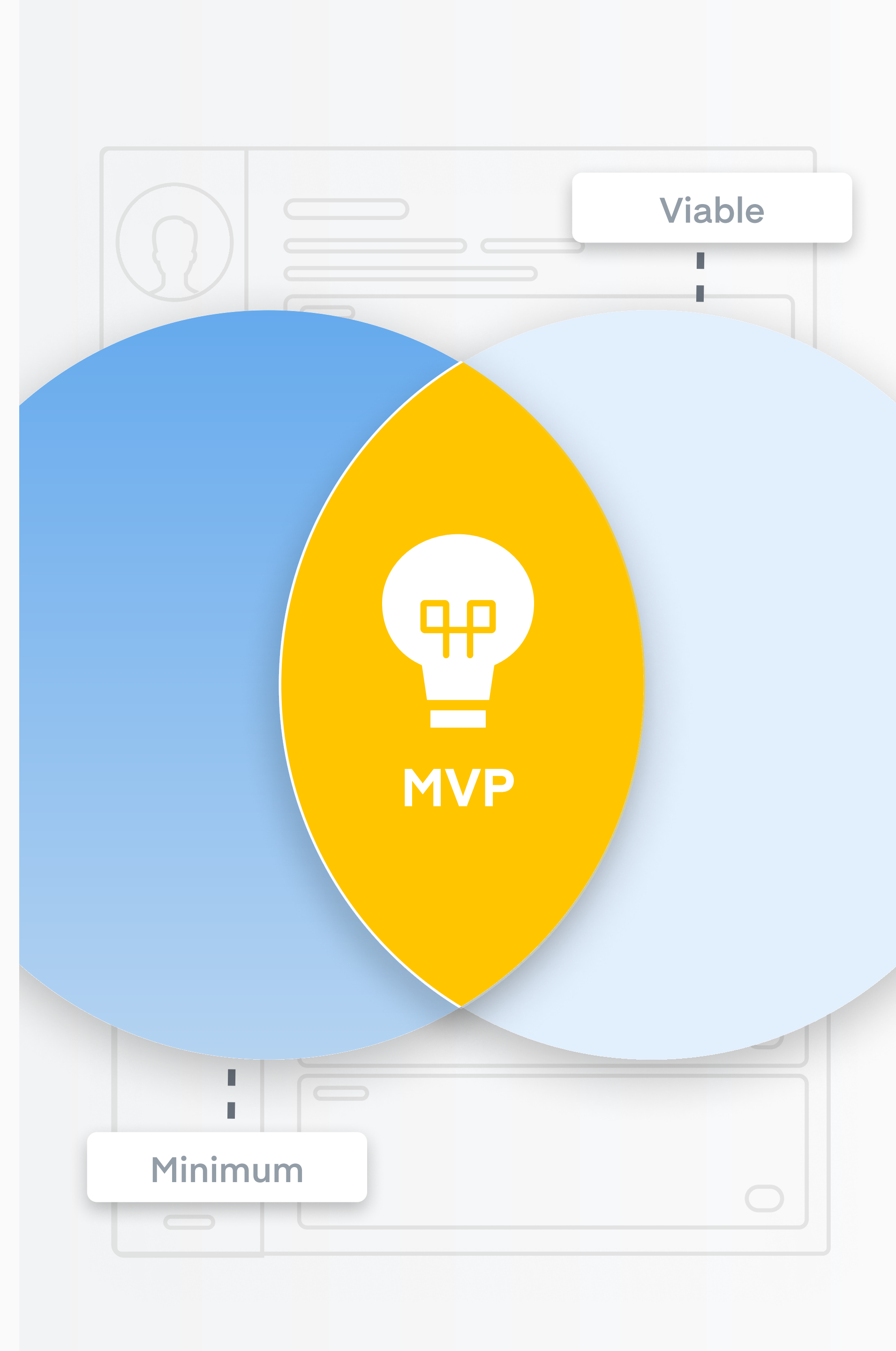


# How to Implement Product Frameworks: Minimum Viable Product

A step-by-step guide to setting up your org and processes around the MVP framework



# Defining the MVP Framework

According to [McKinsey](#), 50% of launches don't hit their targets. Product failure feeling like a literal coin toss is especially pressing for early-stage companies and lean startups that are expected to do more with less.

There is more urgency in launching products at pace that earn enough revenue to keep the business afloat. This is why the minimum viable product (MVP) framework is so appealing to these strapped teams looking to move quickly.

Unfortunately, "minimum" becomes the operative word that can lead many product teams astray. Minimum should not be synonymous with low quality.

**Minimum refers to the amount of investment—time, money, and resources—you need to ultimately discern if a consumer will pay for what you plan on building.**

To make sure your next product or major new feature launch actually drives growth, the MVP framework can be leveraged at lean startups to avoid risk upfront. This requires product teams to validate their assumptions before they build their MVP. When shipping within the context of an existing product or product portfolio, the best MVPs are limited in scope (e.g., breadth, depth of functionality) but still high quality (e.g., reliable, bug-free, secure, reasonably performant). While the MVP framework arose to help lean

startups, it has since been applied to launching new products or major new features within established organizations.

In a nutshell, the MVP framework only works if your product team is leveraging actionable customer data, collecting the maximum amount of validated customer insights with the minimum amount of effort. It's not just about shipping something fast's about customers wanting to:

- Initially purchase your MVP
- Return to use it again—bonus points if these are because they are adopting what you believe are you product's most powerful features (aka the "north star metric")

# When to Use the MVP Framework

- You're introducing a brand new product or major new feature that will require substantial investment and carries many risks
- You need the shortest route to deliver value to your early adopters / first customers quickly so that you can generate learnings
- You have a more granular feature in mind that has to be validated early via a prototype
- You're a lean startup or a team that otherwise moves fast and works with tight budgets

## ✓ PROS

### Mitigate Bottom Line Risk:

Reduce wasted resources and effort with shorter dev times that test ideas "in the wild" sooner

### Validate Value:

Feel more confident that what you're shipping to market will satisfy current customers

### Scale:

Use feedback to improve your product over time — no need to focus on advanced features that customers don't immediately need

## ✗ CONS

Some may misunderstand MVP by shipping something low quality

In some environments, especially in highly regulated markets, it is difficult to skip over the lengthy process of R&D to rapidly bring something to market for validation

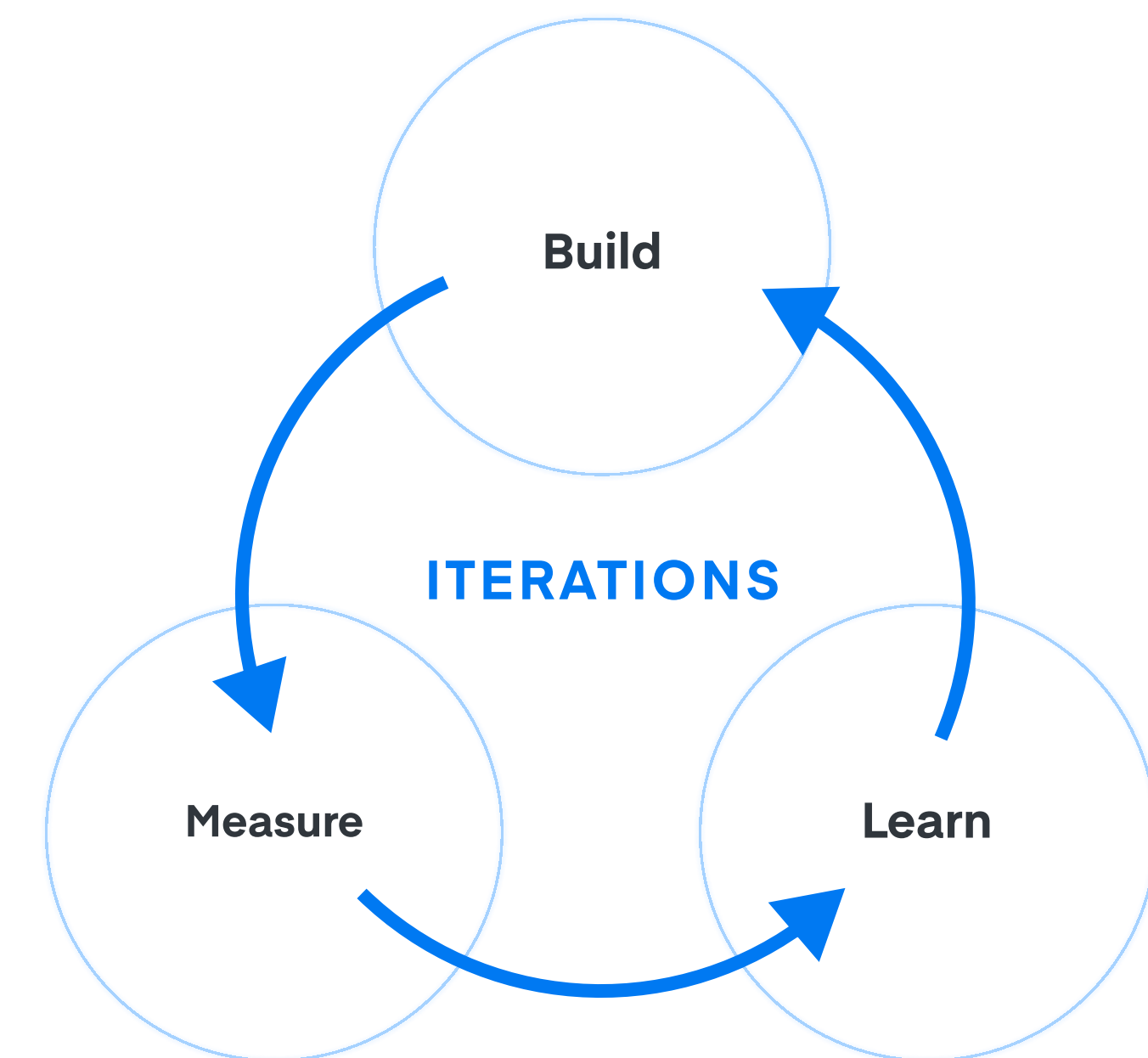
# Anatomy of a Minimum Viable Product

An MVP is a version of a product with just enough features to satisfy early customers and provide feedback for future development. A typical MVP may consist of the following components.

## COMPONENTS

- **Core Features**  
Identify the essential features that address the primary needs of your target audience. These features should be the core functionality of your product.
- **User Experience (UX)**  
Don't let great be the enemy of good here. You want to strike a balance with a UX that's "just good enough" to help you get an accurate reading on whether your product is really creating value. Too much time spent on UX is potentially wasted optimizing features that customers might not actually use anyway; too little investment could muddy the waters and make it unclear whether customers didn't need the features or couldn't figure out how to use them.
- **Scalability Considerations**  
Design the MVP with scalability in mind. While it may start small, the architecture should be capable of handling increased user loads and additional features in the future.
- **Performance Optimization**  
Ensure that the MVP performs well and meets acceptable performance standards. Users should have a smooth experience without significant delays or errors.
- **Feedback Loops & Analytics**  
Implement a feedback system to collect input from early users. This could be in the form of surveys, feedback forms, or direct communication channels. Incorporate basic analytics tools to track user behavior and gather insights. This data can be valuable for making informed decisions about future feature development.
- **Reliability and Stability**  
Although it's a minimal version, the product should be stable and reliable. Users need to trust the basic functionality provided.

All this should allow you to test your product in the real market, gather feedback, and iterate on subsequent versions based on user responses and evolving requirements.



# Your Step-By-Step Implementation Guide

Let's dig into how to successfully achieve each step so you can move onto the next.

\* We'll use an overarching scenario as an example: You work for a new rideshare app company, and you are looking to generate learnings from your launch.

## 1 Identify customer pain points

First things first, you must review your customer feedback and conduct a broader market analysis. Your customers will provide valuable insights into what is working well and what needs improvement.

**Identify patterns to help prioritization:** Actively engage with customers through surveys, interviews, or feedback forms. Look for patterns and common themes in their comments to identify the most pressing pain points. If one or two comments have been echoed by multiple users, your MVP should focus on solving that pain. But don't forget the importance of context. It's one thing for people to tell you what their needs are... it's another if they are telling you that they have gone out of their way to develop their own hacked together solution for that need. Now that's a real need they might pay you to solve!

**Look beyond immediate competitors:** Market research helps you understand the broader industry landscape, competitor offerings, and emerging trends. It complements customer feedback by providing a more comprehensive view of potential opportunities and challenges. Use a combination of quantitative and qualitative research methods, and explore adjacent industries for inspiration.



Feedback reveals that users from competitive rideshare services are dissatisfied with long wait times for rides and lack of transparent pricing. Your team conducts market research and discovers that potential users in a specific city (Minneapolis) face challenges with unreliable transportation options, leading to a significant demand for a more efficient and transparent rideshare service

## 2 Create a product requirements document (PRD)

A PRD serves as a blueprint for your MVP, outlining its features, functionality, and specifications. It provides a clear guide for development teams and ensures alignment with the overall vision. At a minimum, a PRD should include the following:

- **User persona(s):** Who are you solving the problem for?
- **Outcome(s):** What problem will the MVP solve?
- **Scope & release details:** What absolutely needs to be included in the first iteration? What are the essentials that can be separated from the nice-to-haves in your roadmap?
- **Design & wireframes:** What's the big-picture view?
- **Risks:** What uncertainties do you have, and what's the likelihood of the event occurring?
- **Constraints:** What are the restrictions to your MVP? Can it be solved for in a later iteration—and does it need to be?
- **Dependencies:** What needs to be done before another task can be completed? How can we minimize dependencies as much as possible?

**Focus on the essentials:** Clearly define the core features essential for the MVP. Prioritize features based on their impact on solving customer pain points (as mentioned in Step 1). Keep the document concise, and don't feel compelled to deliver everything in the scope.

**Visualize user flow:** The more you can use graphics (e.g., customer journeys, process flows, wireframes, etc.), the more context you can provide for your solution. The wireframes don't have to be perfect or visually stunning; it's more important that they convey the overall idea of the solution and how end-users will flow through the product.

\* You create a PRD outlining the following essential features: real-time ride tracking, transparent pricing, and a user-friendly interface.

### 3 Create a roadmap

Outline the planned development and release milestones. It helps prioritize features for future iterations, providing a strategic vision for the product's evolution.

**Remember the "minimum" in minimum viable product:** Break down the roadmap into achievable milestones. When planning the sequence of your releases, remember that the first iteration that gets tested in the market should be "just enough" to satisfy your users—that could mean solving for the biggest pain first, and then saving more cutting-edge features for a future release.

\* The initial focus for your roadmap is on providing real-time ride tracking and flat-rate upfront pricing across three major cities: NYC, LA, and the currently underserved Minneapolis. Since the focus for the first iteration is to quickly deploy in Minneapolis to capture that market, the roadmap includes a phased rollout of additional features such as driver ratings and loyalty rewards.

### 4 Test for validity

Ensure that your MVP aligns with customer needs and expectations. This step is essential in understanding whether the chosen features effectively address the identified pain points.

**Expect iterations:** Conduct separate problem interviews and solution interviews. This is to make sure you understand the real need, and that you've identified the optimal solution to those needs. This will answer the big question of, "Does my product / feature do something valuable for users?". Usability testing comes later to observe how real users interact with the MVP.

This will answer more minute questions like, "Is the interface intuitive to use?" Pay attention to user feedback and iterate on the design and functionality based on the insights gathered.

\* Usability testing is conducted with a small group of beta users who confirm that the app addresses their pain points by providing reliable and transparent rides.

### 5 Build and launch

Now that you have validated that the MVP is something a consumer would be willing to purchase, the time has come to get this to market!

**Minimize time to market:** Use agile development methodologies (e.g., Kanban, Scrum, XP, etc.) for flexibility—and to allow for quick iterations. Speed and simplicity are key when building your MVP. Focus on the core features identified in the PRD and nothing more.

\* The rideshare app is developed with the core features and launched in those three cities. Your marketing team actively promotes the app, offering incentives to early users. The app's simplicity and efficiency contribute to a successful launch.

## 6 Measure success and note learnings

Just because your MVP is in the market doesn't mean the product is complete. It's time to analyze your key performance indicators (KPIs) and user feedback to assess the impact of the MVP. Learning from the launch informs future iterations and improvements.

**Consolidate feedback:** Make sure your ongoing user feedback is collected and centralized in one place. This will make it easier to have a holistic view of all your potential improvement areas and, more importantly, make sure nothing is missed. To support quick iterations, product teams need easy access to all feedback forms and requests—if you use many tools to collect feedback, such as Zendesk and Gong, it's important to see if you can integrate these scattered repositories into one hub before the feedback starts pouring in. This is especially important if your organization has data silos; for example, Customer Success may be the only team with permissions and access to Zendesk.

**Patience is key:** Keep in mind that a successful MVP doesn't automatically ensure long-term success. The objective is to reduce risks and establish confidence in your development process as you work towards creating a product that truly resonates with customers. It's common to iterate multiple times, incorporating new data and adjusting assumptions to enhance the product's fit with market needs.



The rideshare company monitors KPIs such as user acquisition, average wait time, geographic coverage, and user feedback. Positive feedback indicates that the app is generally addressing users' needs and they are happy with the short wait times, while data on geographic coverage highlights areas for improvement. Your product team notes these learnings for future iterations and shares this feedback with the recruitment team so they can hire more drivers to serve the neighborhoods with less coverage.

## 7 Change management

Implementing any new framework requires that all internal stakeholders are prepared for and aligned with the new direction—not just product teams.

**Determine the need:** Before anything, you must assess the impact of the proposed change on the people, process, and technology. Start by understanding where improvements can be made. Have one-on-one discussions with team members and ask about their work to answer questions like *“What's working well?”*, *“Where do they face challenges?”*, and *“How have you tackled problems before?”*. Don't reinvent the wheel; people often find effective solutions even without fancy tools, so make sure to uncover existing solutions before making any major investments.

**Use the WIIFM communications approach:** The “what's in it for me” style of communication enforces value for the change. Map out your stakeholder groups impacted by the change and assess what they will be most interested in. When introducing the change to their stakeholder group, leadership should always start with how this will benefit them.

**Have a top-down approach:** Change starts at the top. Executive leadership should communicate the rationale for the process change clearly and passionately.

## Follow the 5 D's

To plan on how change will be effectively rolled out and managed long-term, you must Designate, Determine, Decide, Develop, and Decrease. (For more information on the 5 Ds, watch this [change management webinar](#) between Productboard and Joshua Childs, Director of Product Management Email Solutions at Validity.)



## 8

**Build your change community:** Change starts at the top. While executive leadership should communicate the rationale for the process change clearly and passionately, there should also be change champions that help promote change across the company culture. When it comes to picking change champions, a complementary mix is key.

- **The educator:** They know the framework very well and can help quickly align those who don't, creating performance milestones to keep everyone excited and on track
- **The project coordinator:** They understand what the vision is for the new framework, and with their decent amount of exposure to the rest of the organization, they hold the right people accountable and routinely check in on output and status
- **The team peer:** They are closely tied to the group you want to adopt change, so it's important that they really understand the vision and the framework to become an influencer for the rest of the group

**Provide (continuous) training and documentation:** Supporting the transition will not be a one-off conversation. Put new / updated processes, goals, and KPIs in writing.

# Ways to Measure Success

As previously mentioned, the best way to tell if your MVP is successful is if people show willingness to buy it. Here's a checklist of all the metrics you can assess:

## Market Value & Demand Metrics

### User Engagement

- Active users
- Session duration

### Usage & Retention

- Retention rate
- Churn rate

### Usage Satisfaction

- Customer Satisfaction Score (CSAT)
- Net Promoter Score (NPS)

### Financial

- Revenue
- Average Revenue Per User (ARPU)

### User Acquisition

- Conversion rate
- Cost per Acquisition (CPA)

## Quality Metrics

### Performance

- Load time
- Error rate

### Scalability

- Scalability index (ratio that compares change in performance to change in user load)

### Feature-Specific

- Feature adoption rate
- Drop-off rates

## Internal Process Metrics

### Resources

- Development time
- Headcount /  
manpower needed
- Speed to market

### Learning & Iteration

- Time to collect, analyze  
& implement feedback
- Iteration speed

Some of the most successful products started out as an MVP. Dropbox is one famous example. A leading platform in providing secure file sharing, collaboration, and storage solutions, you've probably used it yourself. To become the popular corporate tool that is today, Dropbox followed the MVP framework to launch a minimum viable product to their high-tech adopters first.

Instead of building a complete solution that would require overcoming extreme technical hurdles and months of dev time, co-founder Drew Houston created a simple three-minute video demonstration of what the technology would be capable of (and how easy it was to use).

Houston successfully validated his hypothesis that people wanted this type of file-sharing software. This simple video (the MVP) led to 75K people signing up to wait for a beta invite. The rest is history.

# Helpful Tooling

When launching an inaugural product or introducing a brand new feature into your existing suite of offerings, your business wants to be on the winning side of that 50/50 coin toss mentioned earlier.

There's a strong need to shift from occasional product launches to a faster and well-planned delivery to the market, especially in today's remote work setups where teams are spread out. This requires updating the way we manage our product roadmaps to be as transparent and strategic as possible, so every team member knows what they need to be doing and when. **You need a product management tool that can support the rapid iterations that the MVP framework relies on.**

[Productboard](#) is a product management platform designed to help product teams centralize product-related information, prioritize features, and make data-driven decisions throughout the product development lifecycle. Here's how Productboard can assist product teams in navigating this framework at scale—no matter how small of a startup you may be:

Collect the maximum amount of validated customer insights with the minimum amount of effort. And do it more efficiently than your competitors. [Try Productboard for free to see how we help you get the right products to market, faster.](#)

Productboard can assist product teams in navigating this framework at scale—no matter how small of a startup you may be:

## WITHOUT PRODUCTBOARD

Historically, customer feedback was scattered across many tools, some of which product managers didn't even have access to. Finding the gatekeepers of the data adds time and strain to the process.

Product managers often find themselves spending hours, or even days, on manual tasks. These tasks involve searching for and reading notes scattered across various platforms and Google Docs. This fragmented process consumes valuable time and creates challenges when attempting to analyze feedback and identify emerging trends.

Lacking a unified platform that actively involves your customers throughout the product lifecycle makes it challenging to validate your assumptions before incorporating them into the MVP. Imagine investing substantial time in developing wireframes and conducting user testing on that wrong problem.

When roadmaps are not based on relevant customer data, organizations lose stakeholder trust when the MVP fails. Misalignments ran rampant between the product team and business goals—not knowing who is working on what, uncertainty over the next stage, and dependencies being overlooked, to name a few.

## WITH productboard

### Leverage actionable consumer data:

Collect input from various sources (customer feedback, internal stakeholders, market research, etc.) with [Insights](#). Productboard serves as your centralized repository so you can identify critical customer needs and emerging opportunities. This ensures you prioritize the right features and initiatives.

### Hone in on the essentials:

Define, document, and prioritize detailed product features and requirements with the [Features board](#). Outline specifications, user stories, and acceptance criteria, ensuring a clear and shared understanding of what needs to be included in your product brief or PRD.

### Easily validate ideas and collect more feedback:

Share plans, gather feedback, source new requests, and announce launches with the public [Portal](#). Invite everyone who requested a feature or new product to join a beta of your MVP.

### Ship quickly, but strategically:

Save your product managers valuable time with the [Roadmap](#). Productboard allows teams to create and visualize product roadmaps, supporting the rollout of your MVP from the first iteration onward. Plan toward key milestones, track dependencies, and monitor progress to help teams stay focused and reduce the risk of wasted effort and costly delays.



## About Productboard

Productboard is the customer-centric product management platform that helps teams get the right products to market faster. Over 6,000 companies, including Microsoft, 1-800-Contacts, and UiPath, use Productboard to understand what users need, prioritize what to build next, and rally everyone around their roadmap. With offices in San Francisco, Prague, and Vancouver, Productboard is backed by leading investors like Tiger Global Management, Index Ventures, Kleiner Perkins, Sequoia Capital, and Bessemer Venture Partners.

Learn more at [productboard.com](https://productboard.com) and follow [@productboard](https://twitter.com/productboard).

