

How to set up a Minimum Viable Product (MVP)

Medium.com/innovation-machine/from-a-problem-to-a-well-defined-mvp-in-8-steps-88efa12c1345

August 27, 2020

This article provides **actionable guidance** on how to define a good Minimum Viable Product and avoid common mistakes and risks. If you are in the process of designing a product, or setting up your startup or otherwise involved in product management, read on.

1. Start by framing the problem

One of the most important steps in the product development process — is the **understanding and proper articulation of the problem**. It is a good idea to **use a model, a structure** to help you formulate the problem with clarity.

Start by describing **the ideal situation** versus **the current one**, and **how certain users are impacted** by this gap

Support your problem statement with *statistics and facts* — for example, figures to describe the **size of the problem** or the **potential of the opportunity**. Keep it simple and clear- a structured, well-understood articulation of the problem with no technicalities and fancy terms.

The problem statement should become part of your ‘*common corporate language*’: your team, your investors, your sponsors, and other stakeholders should all be able to instantly understand it and reference it when reviewing your solutions and product plans.

2. Identify your users

Make sure you know who you are solving for! Identify and *name* the different classes of users in the context of your problem; document their needs and the problems they are experiencing; identify their pain points; their expectations, and the best possible experience they could have in this context.

Define Success criteria for each class of users.

3. Understand your users

Identifying your users is different from *understanding* your users. You need to apply empathy, study your users and deeply understand their profiles, habits, and needs. You

could use existing research and public domain insights; or host user interviews and focus groups — to validate both your problem statement and your potential solution.

4. Validate the problem

Having your problem defined with clarity, allows you to validate it with your key stakeholders — including your customers or end-users: challenge the problem and make sure it is *worth-solving*; study the impacted users and document how they are affected by this problem. During this process, try to answer the following questions:

1. Do users know any *potential solutions* to the problem?
2. Is this the most significant problem they are facing— in this context?
3. Are there any workarounds they are using?
4. How would their situation improve if there was a good solution?
5. Would they pay for such a solution?

At this point, you must also scan the market and the state-of-the-art to figure out if there are existing products or services — addressing the same problem; and if so, it is critical to understand how.

5. Ideate on potential solutions

Having a *well-defined* and *validated* problem enables you to ideate and explore potential solutions. At this stage, I would recommend starting by setting the context — make sure your team is aligned and has a deep, shared understanding of the problem space — the situation, the ideal state, the users, the personas; the pain points and the opportunity.

Then, move on to an ideation phase — *you need ideas on how to solve the problem and provide value to your customers*. Ideation could take the form of a series of brainstorming sessions, or **design sprints** or internal contests like **hackathons**.

Whatever the form or methods you select, make sure your team is capturing all of the ideas, into a system. This is important to allow fast iterations over this set of ideas, and post-processing: you will have to evaluate each idea and attach metadata and artifacts as you go. Depending on the scale of your initiative, **an ideation system** could add significant value by organizing and speeding up the entire process.

Assuming a set of great ideas and potential solutions is there, iterate through the following steps:

1. **Review all your abstract ideas and prioritize them** based on their *potential* and *feasibility*. *Do not discard ideas; assign priorities instead* (this will allow ideas to *stay active* and be reconsidered under different circumstances in the future). Prioritization will allow you to pick the most

promising ideas, using not just opinions but a standardized ‘idea assessment framework’.

2. **Combine ideas** as necessary — merge or group them — to draft an overall solution or **product definition**
3. **Iterate and refine the product draft**; make sure that has the integrity required to call it a ‘product’
4. **Start Small — but Think Big** and define a long-term product road-map
5. **State your assumptions** — and how you are going to validate them

6. Define your ‘full Product’

The *Minimum* in the MVP implies that you already have the *Big Picture*, the *product vision!* A common mistake is when the team ‘easily’ identifies a set of ‘obvious’ use cases as the MVP — without a clear product vision and a good definition of the ‘complete product’.

Assuming you have this bold *product vision*, the next step is to define your ‘complete product’ as a long list of User Stories — your product backlog. It is important to understand here, that this is the *full version of your product* / not just your MVP!

Another important point is that **you don’t have to apply feasibility, cost or other constraints at this stage**; my advice is to describe everything — even the craziest and expensive product features, as you will be able to prioritize and manage them at a later stage.

This way, you don’t have to skip, drop or archive an idea for a feature that looks ‘ahead of its time’ or not well-understood yet. Instead, you should include them in your backlog with a lower priority — but they will still remain discoverable and potentially useful in the right context.

Iterate and keep defining more user stories, until your product is described in full. Your ‘full product’ backlog should have all the features you can think of, reflecting the needs of all users identified so far; and all in the form of solid user stories.

7. Define your MVP — the Minimum Viable Product

At this point, you have the definition of your ‘full product’ — the *complete* not the *minimum*. What you need now, is a process to find the *best minimum subset of features* from the *complete product backlog*.

This ‘*best minimum subset of features*’ which delivers enough value to your early customers to keep them happy and engaged, is what the MVP is all about: the first instance of your real product, which will help you to *go to market faster*, with *minimum implementation costs* and *the right feedback loops enabled*.

To find this minimum subset, analyze carefully each User Story — in terms of *value to the user*, the *importance of solving the problem* and also in terms of cost and feasibility.

This way, **all user stories in your product backlog will get a priority** (a number — ideally as a function of the expected value and feasibility).

The next step is to **rank the User Stories**, with the highest priority at the top; then, you have to apply *business* and *product sense* to draw the red-line which will define the top-n stories as the basis for your MVP.

8. Define how success would look like

By now you have a great basis for building your MVP: you have a solid