

Struggling With Your MVP—Here’s How to Build It Right in 2018

 medium.com/swlh/how-to-build-an-mvp-in-the-right-way-in-2018-f538df0f2bba

Swarnendu De

January 30, 2018



Courtesy: [Innofied Solution](#)

“I have got an idea!” 😊 😊

That’s the foundation of many successful and unsuccessful stories behind a number of products.

Nod your head in **‘NO’** if you have not heard the same story.

It starts with an idea—but how to implement it, and become successful is still a mystery to many. Right?

Taking a look at the Silicon Valley, one will feel immersed in their vibrant entrepreneurial culture. And why only **USA**, startups are mushrooming in various global locations (**UK, India, Pan Asia, Gulf**) where everyone enjoys an exciting and perilous journey to an uncharted territory. I too embarked on a similar journey and from a software engineer to a CTO, and finally turned into a Founder under 30s.

Most of the products I built, reached thousands.



Now when I hear from startups everyday... I see them stuck in a similar wave, in which I was trapped some years back.

They build an MVP, does their marketing to-do's but ALAS! The MVP just didn't work the way they wanted.



What Make These Young Startups Dump Their MVPs?

If a startup has his MVP ready and validated, then what makes him or her fail.

Well....

Bringing a new product to the market is always a BIG risk. But to reduce it, you build an MVP. Despite building MVP you could not minimize the risks due to some gaps.

Here are some strange cases which I witness daily while working with startuppreneurs at Innofied.

#1. They come without doing a proper homework (customer's needs, market research, demographic research etc.) 😞😞😞

You can build an MVP any time simply sharing the wireframes to your technical partner. But that is not a solution unless you do your homework.

Most of the product owners come to us **without doing a proper research on customer's needs**. They come to ask without asking two questions to themselves —

- Why do I need this product? and
- How can it help me?

These answers will help them to entail a better product to future customers. But they never ask.

They also miss on early competitor research. It is obvious to conduct a competitor analysis to know if there are similar products in the market. Even if there is no direct competitor, your faith in the uniqueness of the product is not sufficient to bring it up in the market.

Competitor analysis is necessary or else your MVP is not going to have the impact you wish to have.

A Step-by-Step Guide to Conducting Competitive Analysis

"Never interrupt your enemy when he is making a mistake." -Napoleon Bonaparte When your competitors make mistakes, it...conversionxl.com

#2. They spend a lot for all platforms, spending above their budget 😞😞

If not all, most startups are bootstrapped or have investors. The angel investor might have faith in the product and agreed to bootstrap the startup. **But having capital does not mean you need to develop products on various platforms.**

Many startups who come to us to build a web or mobile app MVP want to develop it for all platforms. That's a sheer waste.

If your MVP fails, you may not have much capital left to start fresh. It is better to build an MVP for a particular platform and beta test it.

5 Ways Successful Founders Bootstrap--and Key Things They Don't Do

Bootstrapping. Some entrepreneurs do it out of necessity. Without significant capital or willing investors, there's no...www.inc.com

#3. Do not have a clear idea of what features to implement and which should come in the later phase. 😞😞😞

It is important to prioritize the features that you want to implement in your MVP. Before you build an MVP highlight all those features that you think are nice-to-have. But it is not strictly necessary.

Have a list ready to prioritize and ask questions to yourself like -

- Which single action do you want your users to accomplish through the product?
- What more features will I like to add to make the product more efficient in the later phases?

Before asking a developer to work on the MVP it is important to set categories as 'must-have,' 'nice-to-have', and 'don't care'—then ask the developer to act on it. This will reduce the chance of your MVP to fail.

How 'Minimum' Should Be Your MVP?

If you too are a Startup with some 'billion dollar' app ideas, you may have heard the term **Minimum Viable Product**. You know much about it. You have been regular patrons of the

Czars of Startups, gave a read to their thoughts regularly, but what you missed is the fact —*how minimum your MVP should be.*

Well, a real MVP should deliver minimum yet a valuable product, and it must-

- 💡 Serve at least one specific audience
- 💡 Address at least one key problem
- 💡 Have a well-designed user experience
- 💡 Be easy to build and launch quickly

If your MVP is not delivering these features—then it is not minimum and neither viable or valuable.

“...MVP is not about building a Beta version of your app. Neither creating few workable mock-ups. Implementing the minimum features, without the essence of its viability, is definitely not the MVP.”

You definitely do not want your MVP to see the doom’s day. Then what saves your MVP from sinking?

Well, what you just need is the **right spice** that will be accepted by the niche you target. A product which is not yet built but has a strong appeal and claims to give access to features that can make the life of users easy.

Say a maximum of 5 features is good to add in your MVP. Then you test, re-run and keep on iterating unless you fix all the bugs required. You want to gift your user a perfect product with minimum errors and to do so, start with the least. Give them least features and if they find it as an addiction—then you’re very near to hit the **GREEN** button!

How to Stop Building Half-Baked MVPs?

Let me be direct and share some of the ideas that will stop you from committing the old mistakes, while you build an MVP in 2018.

1. Invalidated MVP should not see the Doom’s Day so soon

It is important to validate your MVP, as this is the classic idea. What you do while validating an idea is approaching customers and try to sell the product which you haven’t built yet. If they pay you to deliver a solution a month from now, you validate your idea. (👁️)

But what if you fail? Will it invalidate your business model?

A strange case I must cite is that of Dropbox—Drew Houston gathered 70K signups in 2008 with a **Digg Video** and went long to make his product market-fit. But if he failed it wouldn’t have meant Dropbox won’t work—it would have meant that threshold was higher than expected or

Digg people were not the early adopters or maybe those people got attracted to some other events and stopped sharing the Digg video.

Failure in validating your MVP only invalidates a small way to acquire customers—not the whole business model.

Rather invalidated MVPs are an only possible way to build better products in small batches. But makes them difficult when product quality depends on the scale of use or when low-quality alternatives capsizes the market.

It is better to run an invalidated MVP first and test whether a customer will buy a better product rather than pitching validated MVPs and keep on wondering—*“What if I fail!”*

2. Before you jump on the bandwagon—understand and identify your success criteria

Before you build an MVP you should identify the needs as to why the product should exist. This might be organizational or a customer need that can address the recent gap.

You need to address the long-term goal and write it down if you want to answer the following questions —

A. Why are we doing this project?

B. Check the criteria that will make your product successful

How to chalk out your criteria and needs?—Say if you are running a bakery chain then your target or end goal will be to reduce the time-to-checkout by 25%, have a number of monthly users say 100 thousand, and need to reach a target of say \$1 million every month. This is the way to identify your success criteria.

3. To build the right MVP map outline your user’s journey

If you look into the user journey thoroughly you will see that it divides itself into three parts—user, user actions and story endings.

- **You need to identify the user.** As these people will use your product. It is thus possible to gain more than one category of user. Say if you have a laundry booking app, you will gain both the laundry service seekers and the service providers.
- **Make sure to map the end goal of the user.**
- **You need to identify the actions that the users need to take** in order to reach the end of the story and achieve the goal.

When you plan to build an MVP, you will want to know which user has the most job or focus (this will allow you to stay true to the lean startup MVP methodology). In most of the cases, this will make sense—however you might add some quick value. But there are other high priorities that you need to address, and you may need to focus on a different user.

4. Have the story-map ready before you build an MVP

Story mapping allows you to identify the pain points of the user and gains that a user can achieve. This exercise let you determine the greatest potential and adds more value. Then you will be able to focus on the minimum viable product while adding the less impactful ones to the product roadmap for the late releases.

You just need to do three things to define the user flow or the product roadmap.

- **Write down the actions that user need to take or have taken**
- **Jot all the pain points for every action and brief summaries of the inconvenience caused** while trying hard to complete the action.
- **Note the gains you had and achieve the pain addressed.**

5. Use the product roadmap to prioritize the features before you build an MVP

- Collect a number of opportunity statements with the help of pain-gain maps.
- Finalize and find out what features are important for the initial MVP release from those charts.
- List the user and the particular opportunity statements and provide a breakdown of those features to include them in the roadmap.

Do use the prioritization matrix to make the most impact in relation to the urgency of the feature. You are the one who is going to take the final decision on what should be there in the minimum viable product and what you will keep during the launch of your MMP. (Minimum Marketable Product—which you need to need to build after your MVP is validated and successful)

To make things a little easy—I have a **FREE** worksheet for you. Take a print out, solve your queries and pin it on the board to refer, whenever you feel something is still missing in your MVP development.

[DOWNLOAD WORKSHEET.](#)

You're All Set to Build an MVP

You gathered all the things that you need to build an MVP—from knowing the targeted users to addressing their pain points, kind of action they want to take through your product to increasing customer gains—if all seems to be done—you're all set to build a minimum viable product.

To know some more productive ways to build an MVP in 2018, you can refer to this article —[Updated Ways To Build An MVP App For Your Startup.](#)

ABOUT SWARNENDU

Swarnendu De is the Co-founder of Innofied Solution, an Award Winning Apps, and Game Development company.

Founded in 2012, Innofied is one of the fastest growing Software Development companies headquartered at Kolkata, with an extended R&D center in Pune. It has grown globally with a new client proximity center in Sacramento, California. Swarnendu manages the Technical and Business Development Operations at Innofied. With over 10 years of industry experience, Swarnendu has also authored [Backbone.js Patterns and Best Practices](#), published by Packt Publishing, UK.



He is also the Co-founder & CTO of [Crossindia Adventure Tourism](#) and leads their IT and Digital Marketing Operations. He regularly writes at his personal blog, [company blog](#), LinkedIn and other popular platforms.

Follow him on Twitter at [@swarnendude](#).

This story is published in The Startup, Medium's largest entrepreneurship publication followed by 291,182+ people.

Subscribe to receive our top stories here.
