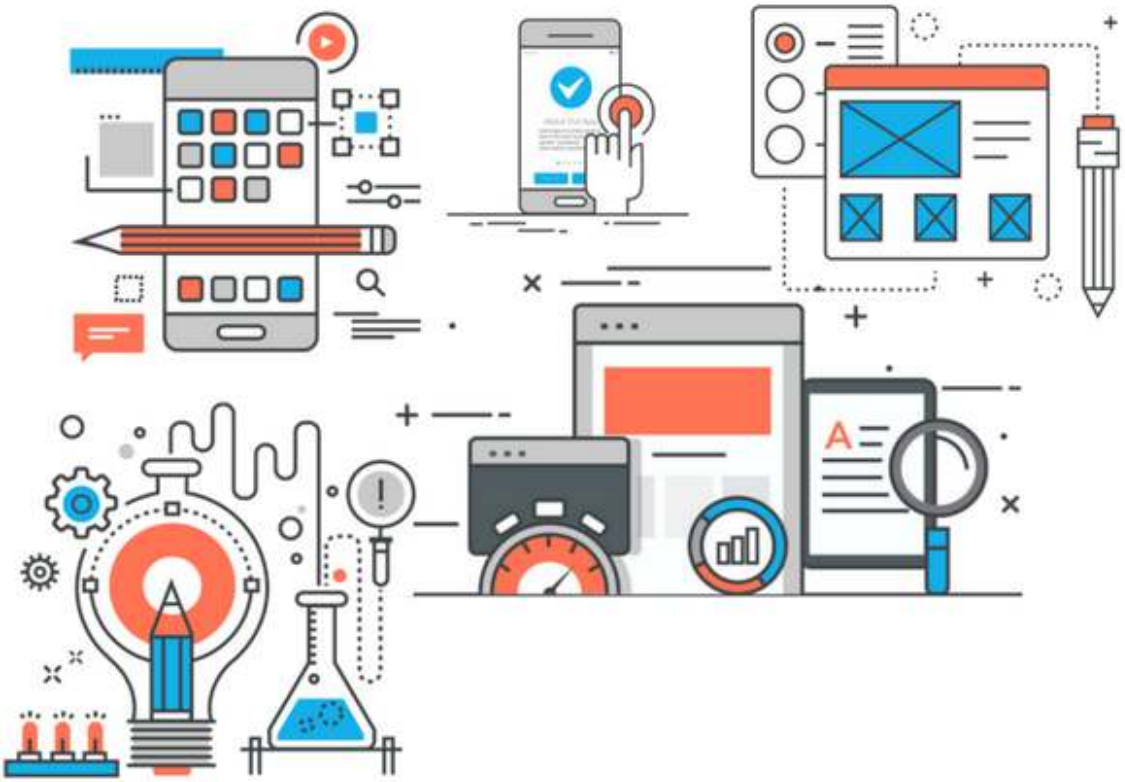


Software development

step by step

neoteric

Table of contents

“Short cuts make long delays”

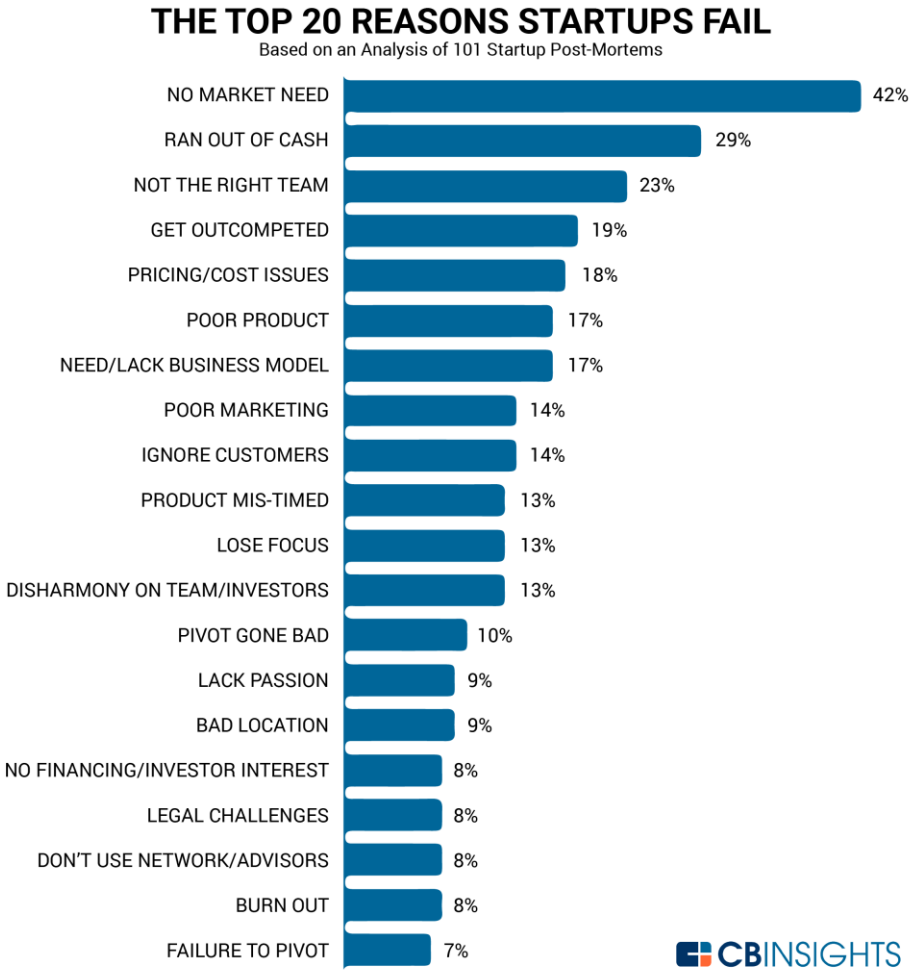
Pippin

The path that leads to having a successful software can be rough and bumpy. It can take a few weeks or long months. The way it goes depends on the product itself - on what functionalities it requires and how complex they are - but also on how you approach it.

Between having an idea and having a working product, there are some steps to follow and milestones to achieve. First, you need to validate your idea and build a proof of concept. Then, you collect feedback and build a prototype - the first version of your product that actually works. Then you gather data, collect feedback, and introduce improvements to get to an MVP. And the path does not end there!

Taking a shortcut seems to be a smart move. Why would you waste your time on building a proof of concept when you could build a prototype straight away? Or show your precious idea to the public before they can buy the product?

Nothing could be further from the truth. According to CB insights, 42% of startup failures are caused by a lack of market need. And that is something that could have been validated before they started building their products - risking their time, energy, and money.



Software development is a complex process. Every step of this process brings some value that you don't want to miss. Let's see what these steps are and what challenges they entail.

Milestones of software product development

Feasibility study (concept evaluation)

It's the first stage of software development. You have an idea. It is absolutely brilliant! And it's sure to change the world. Or, at least, the way people do some certain thing.

But if you think that you only need to do some coding and make the app, you have a lot of catching up to do. I recommend you start here: Things to do before you make your app - 10 successful Polish startups share their advice. Or there: Do not start making your app with an MVP!

What you should actually do now is see if your idea makes sense and if there are people out there who think the same.

You don't need to hire developers at this stage - you usually start doing this alone. You need to dig into the problem that you'd like to

solve, access experts and people who know this problem best, and get as much information as you can. Try to find out:

- What's their problem?
- What's their "win"? What would they gain if they could solve it?
- How do they solve it now?
- What would have to happen for them to change this current solution?

You need to evaluate the current solutions and look for your edge. In what aspect can you be the best? How can you solve the problem in a way that thousands of people will find better than what they have now?

> In order to succeed, you should head for being 10x better and 5x cheaper than their first choice.

At this stage, you build a hypothesis and research for answers. You build your Value Proposition Canvas and Business Model Canvas based on assumptions, then you test them by reaching out to target customers, key partners, influencers and asking if what you assumed is right. You need to go through market reports, keyword and trends analyses, social media reports and analysis, verify your market size, learn where you can find customers and what they engage in. There is a lot of research to be done here!

Proof of concept

You've done the research, you've learned a lot making the feasibility study. You know what you need to do. You thought everything through and there is nothing that can surprise you. Now it's time to show the world that you can actually make your idea come true.

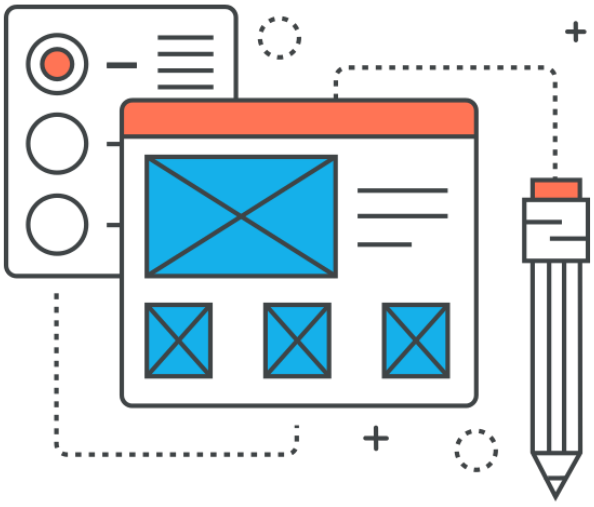
In order to build a proof of concept, you need to create the smallest part of your product vision that makes the change to its target group. It doesn't have to be well designed or have great UX. It doesn't need these cool features that you thought of when you were talking to that guy at the last conference. It doesn't need to integrate with this popular app. It has to be a proof that you can deliver. It can have various forms: white papers, mock-ups, visualizations, pre-miniaturization hardware, key algorithm, or a mathematical proof for a hypothesis.

A proof of concept is made to show that you have a problem/solution fit, to convince people that you can make a change.

Once you have it, you need to get back to your target group and ask if this is something that they would choose over the current solution. If yes, make sure that you understand why - this is your product's unique value proposition.

> At the early stage, feedback is the most valuable currency.

Getting information from potential users will help you develop this product, and avoid making unnecessary features. Moreover, it will be much easier to get influencers and media involved if you already know them and they already know your product. If you implement their advice at the early stages, they will be much more likely to relate to your product. But most importantly, it should convince your future co-founders, partners, experts, potential customers, influencers and early investors (including family, friends, and fools) that you have identified the right problems in a niche and that you have a solution that is a game changer.



Prototype

A prototype is the first working version of your product. It is its simple version that instantly brings your users to the magic moment, which is when they recognize the delivered value. The purpose of this stage is to find users that can be convinced they will benefit from your product, and be able to close the deal.

You should now show your product to the wider public. Exhibit at expos, get first test users, engage with media experts, and watch its reception. This is the time when you can test if the market is actually interested in what you offer.

Just like the previous stage (Proof of Concept), the phase of a prototype is when you should gather a lot of feedback. It's still a valuable currency. What's important, though, is that instead of the declarative information, you start getting behavioral data.

With a prototype, you can observe the users, see how they react and interact with the product. And, of course, you can still talk to them and ask if this product solves their problems, and how it is better than what they've had so far.

The last thing you usually do in this phase is introducing the necessary improvements. You don't do it once, you don't do it twice. You keep going until you have a version of your product that will be good enough for your model customer to say "I want that badly". This is when you're ready to go to the MVP.

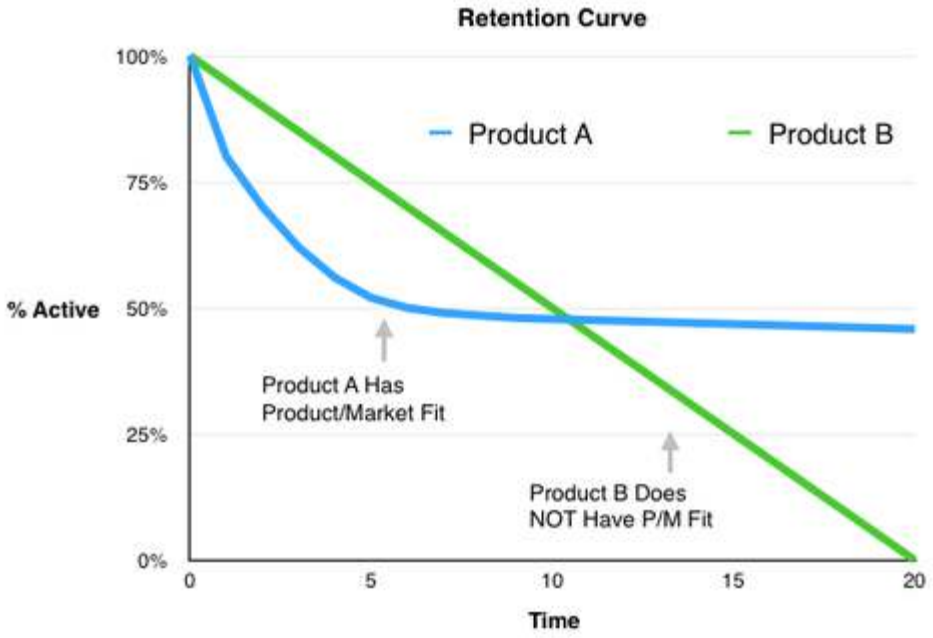
MVP

That's an acid test of your product. This is the last version of your prototype. It's is when the market decides that your product solves an important problem - and is ready to pay for it.

Based on the feedback gained through the earlier stages, you should have analyzed user behavior, verified your hypothesis and tactics, and optimized your product accordingly. You should now be able to prove - with the numbers! - that you are able to scale your sales, access new users in a repetitive way. You should be able to predict revenues and know your basic metrics like ACV, LTV, CAC, time to close, and effective lead generation with the right KPIs.

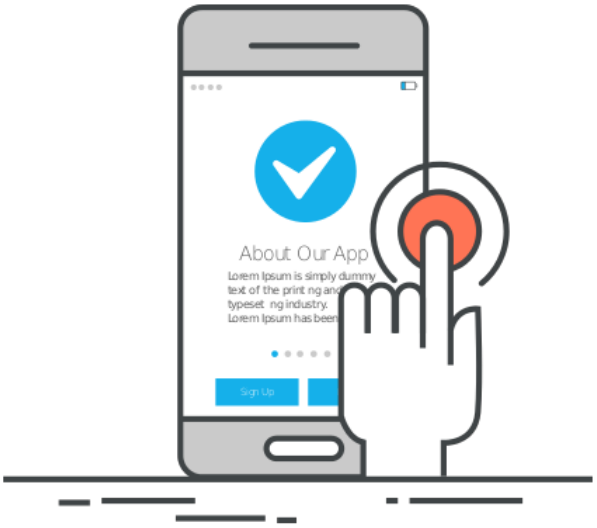
It's okay if the first sales are not profitable - still, you need that phase. MVPs are not always what you'd like to build or end up with. It can be a food truck if you want a chain of restaurants; it can be an open source library when you're looking at a BOT marketplace – it

needs to validate your market fit, show that you can get early-adopters, evangelists and influencers involved, prove that you can penetrate the market, and that you have the right conversion rates. The most important thing here is to get the data. Your goal is to find the most efficient way to enter the market, penetrate it and conquer it.



https://brianbalfour.com/essays/product-market-fit

The main purpose of the MVP stage is to prove that your product is scalable. And you do it by improving your product, your business model, and your customer acquisition process to get to the point where the only limitation for your growth is the cash you need to scale up.



Fully Featured Product

Since you have traction, the market decided that your product is a solution to some problem and that it's better than other solutions, your new goal is to keep this product alive, develop it further and attract new groups of customers (read more about these groups here: Do not start making your app with an MVP). It's time to make money and show a good ROI to your investors. And consequently head towards fulfilling your vision.

Right now, you need to focus on what makes your product successful and optimize for scaling. In many cases, this means rebuilding or refactoring.

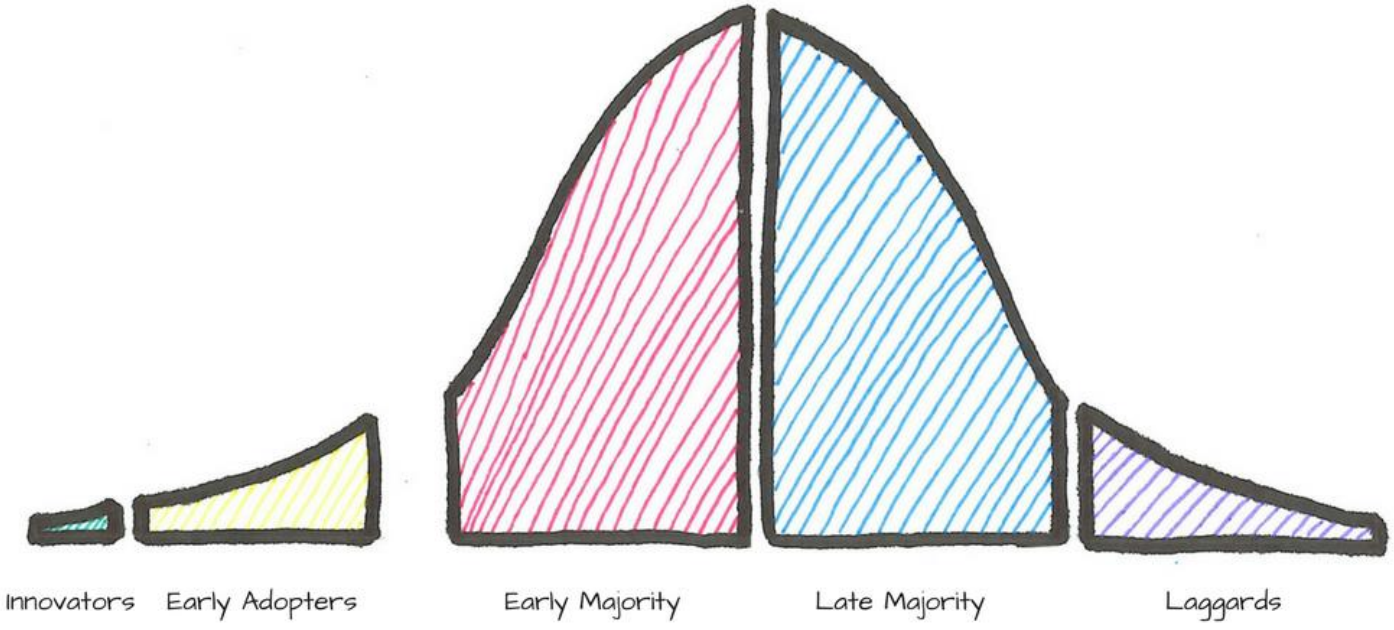
It is the time when you open your first restaurant (and then second, and third) in order to build a chain. It's when you add new versions of your software product, introduce new language versions and adapt your product to new markets, personalize it, extend its capability, introduce new integrations with other tools, and most importantly: find new revenue streams and new markets to expand to.

Diffusion of innovations

Going through all these stages of software development, you are able not only to avoid gaps in your learning curve, but also to target your product to the right groups, depending on the stage.

According to the diffusion of innovations theory, there are 5 types of adopters - depending on their rate of adoption. Those are: innovators, early adopters, early majority, late majority, and laggards.

The division popularized by Everett Rogers, a professor of communication studies, who described the theory in his book *Diffusion of Innovations* (1962), was a synthesis from over 508 diffusion studies across the fields that initially influenced the theory: anthropology, early sociology, rural sociology, education, industrial sociology and medical sociology. Rogers defined an adopter category as a classification of individuals within a social system on the basis of innovativeness.



Each group distinguished by Rogers has different needs, different goals, and different approach to new solutions. In order to succeed with the new product, one needs to know which group to target at each stage.

When you have a/an…	You should target…	Because…
Idea	Innovators	They will be the only ones willing to discuss it. These are the people who are constantly looking for a new way of solving problems, and who are most willing to share their thoughts.
Proof of concept	Innovators	They are most active, most forgiving, and most willing to share feedback. They are ready to overcome some early days problems before your product gets mature and out of bugs.
Prototype	Innovators & early adopters	If they see the value that comes from using your product, they will forgive you the lack of some functionalities or integrations. Even if the product does not satisfy them at the beginning, they are rather likely to give it another try.
MVP	Early majority	They are willing to take a chance with new products (once they hear it's worth to), but they are not willing to take a risk and test the new things themselves before they can read their reviews.
Fully featured product	Late majority & laggards	It's worthless to target them when your product is missing something. They buy a new product when it's not that new anymore. they are not tempted by the exclusiveness of a new solution, and they will not forgive the bugs.

STEPS OF SOFTWARE DEVELOPMENT

Feasibility study

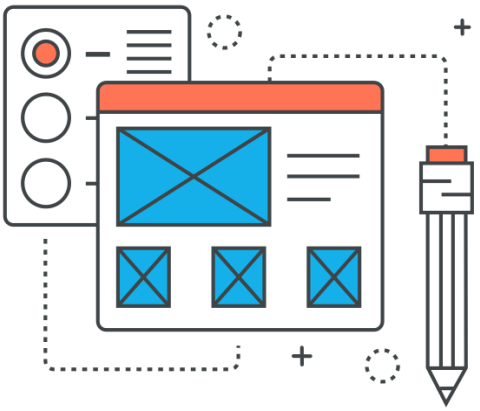
- Take your assumptons and build Value Propositon Canvas and Business Model Canvas.
- Do the research: go through market reports, keyword and trends analysis, and social media reports. Verify your market size, learn where you can find customers and what they engage in.
- Reach out to your target customers, key partners and influencers and try to validate your idea.

Proof-of-concept

- Create the smallest part of your product vision that makes the change.
- Get back to your target group and ask if this is something that they would consider to choose over the current solution.
- Get as much feedback as possible.

Prototype

- Develop your product - add first features, improve the UX. Make it the first working version of your product.
- Show your product to the wider public. Exhibit at expos, get first test users, consult media experts, and watch its recepton.
- Gather feedback and collect data. See how users engage with your product.

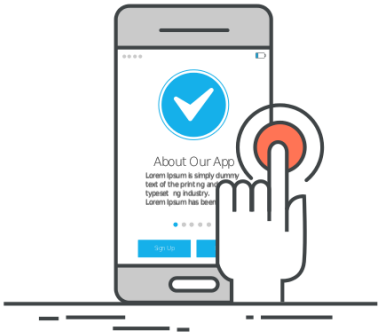


MVP

- Introduce the necessary improvements until you have a version of your product that will be good enough for your model customer to say "I want that badly".
- Prove that you are able to scale your sales, access new users in a repettve way. You should be able to predict revenues and know your basic metrics.
- Get the data. Find the most efficient way to enter the market, penetrate it and conquer it.

Fully featured product

- Keep this product alive and develop it further. Focus on what makes it successful and optmize for scaling.
- Attract new groups of customers: find new revenue streams and new markets to expand to.
- ... keep going!



Good luck with your projects!

Want to learn more about how to build your software product?

> How to validate your startup idea?

> Do not start making your app with an MVP!

> 29 Tools to validate your idea before building an MVP

> Things to know before you make an app – 10 successful Polish startups share their advice

We help startups and enterprises build their products – either from scratch, or by developing existing solutions.

We have your back at every stage of project lifecycle!

Contact us and get an estimate for your project!

contact@neoteric.eu

neoteric.eu

facebook.com/neoteric.eu

twitter.com/NeotericEU