Minimum Viable Product (MVP)

Mahasiswa memahami cara MVP guna menguji ide, mendapatkan umpan balik pengguna, dan meminimalkan risiko kegagalan sejak dini.

Minimum Viable Product (MVP)

What is a
MINIMUM VIABLE PRODUCT

M Minimum

The most rudimentary, bare-bones foundation of the solutionpossible

V Viable

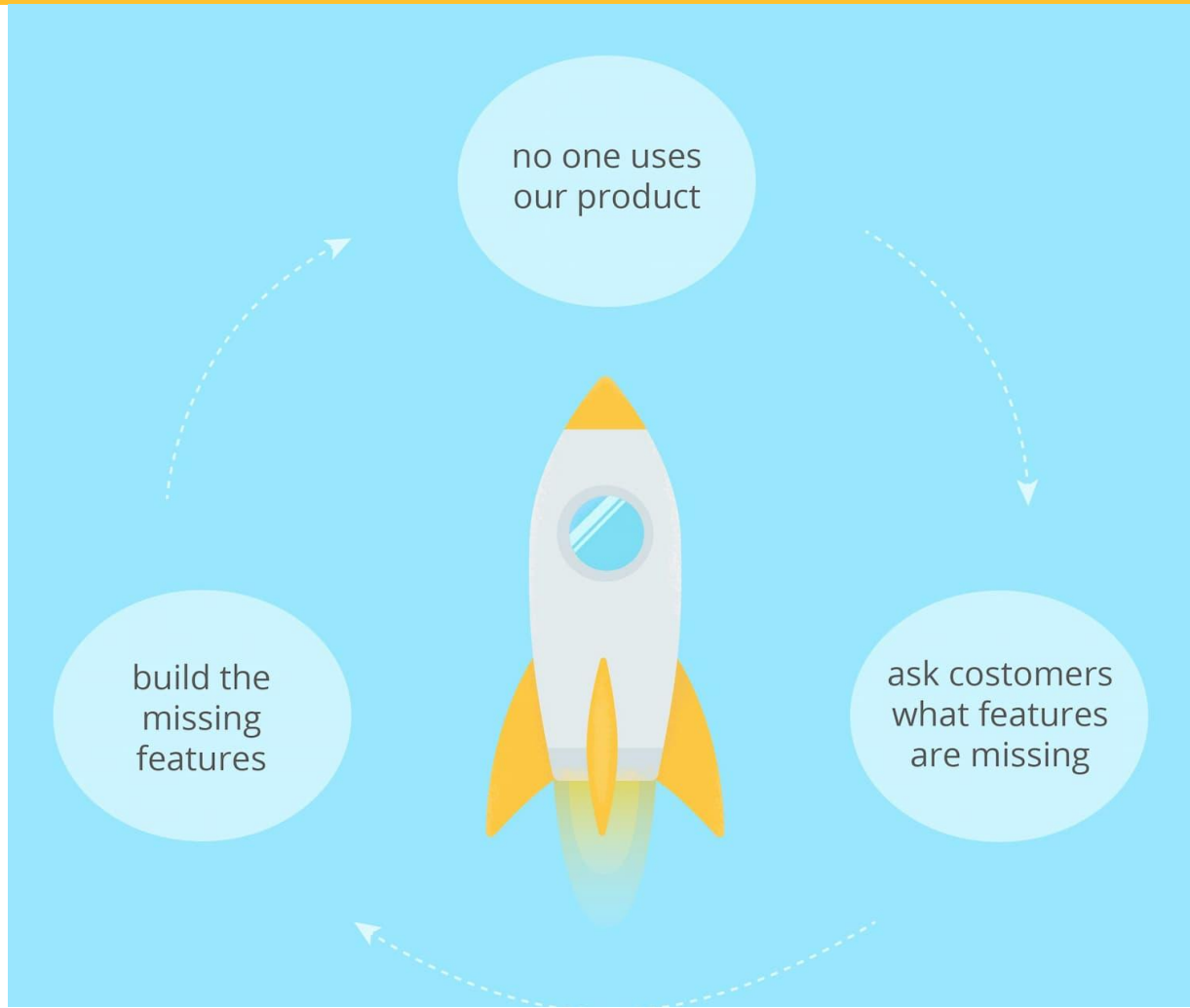
Sufficient enough for early adopters

P Product

Something tangible customers can touch and feel

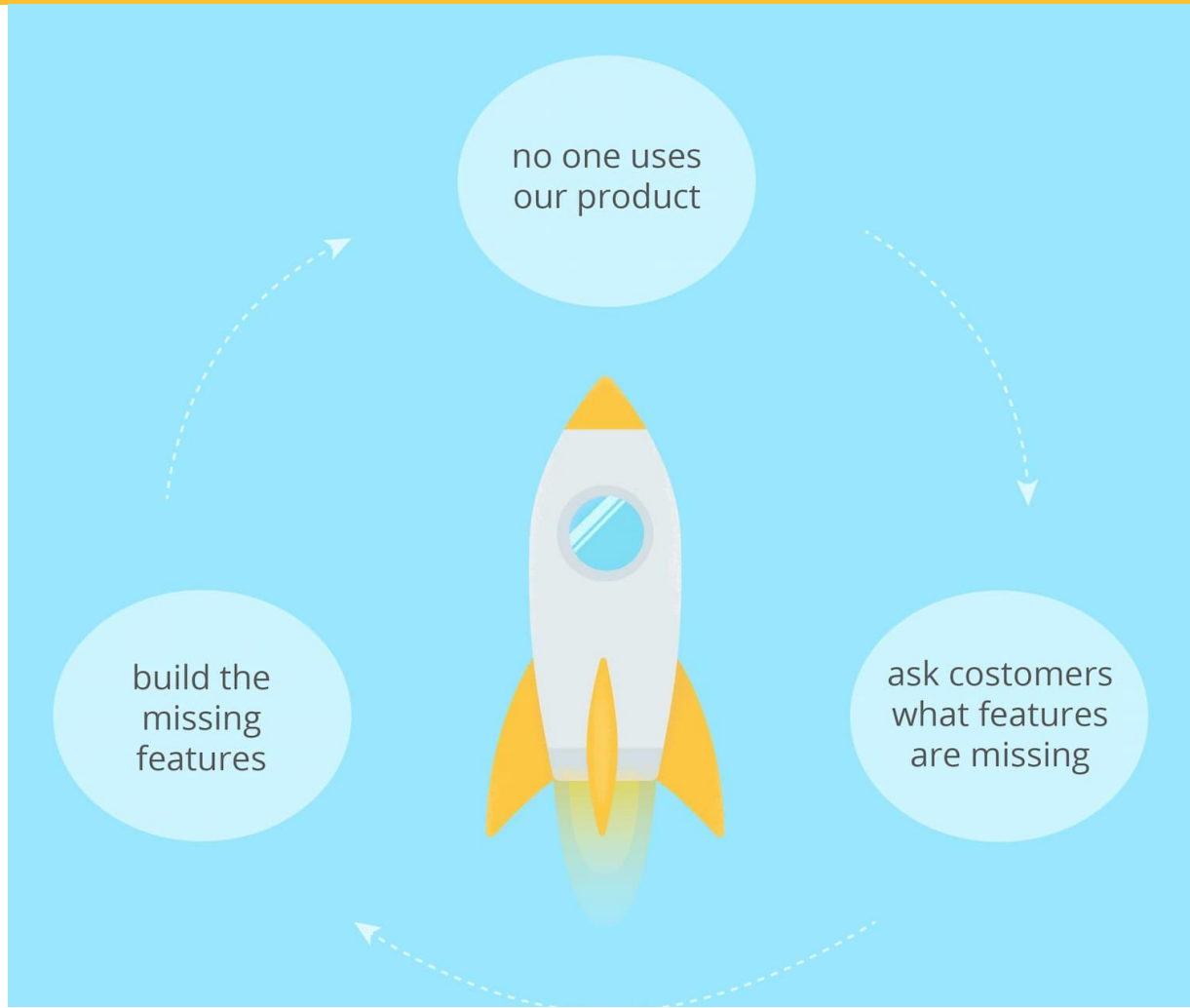
The Purpose of Creating an MVP

- **Focus on Core Idea.**
  MVP helps teams concentrate on a single, clear concept without distractions.
- **Validate Product Early**
  Test your idea with real users before committing time and money to full development.
- **Reduce Risk and Error**
  Catch flaws early and make informed decisions, minimizing costly mistakes.
- **Gather Real User Feedback**
  Understand how users interact with your product and what they truly need.

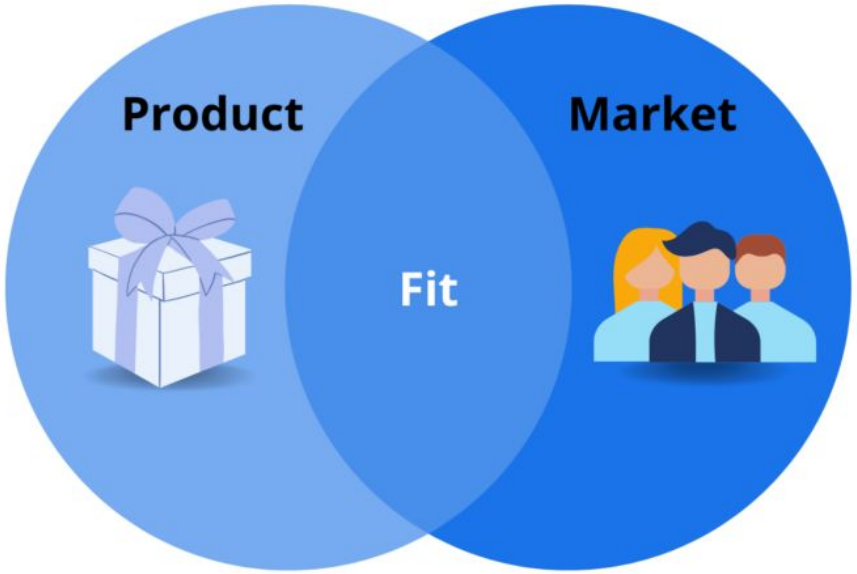


The Purpose of Creating an MVP

- **Guide Product Development**
  Use feedback to shape what features to add or improve for the final version.
- **Accelerate Development Cycle**
  Iterate faster with direct insights from users, reducing time to market.
- **Attract Investors and Early Users**
  A compelling MVP can spark interest, build hype, and secure funding (e.g., Dropbox)
- **Cost-Effective Launch**
  Save resources by building only what's necessary to test your core assumptions.
- **Strategic Growth Planning**
  Identify key areas for future updates and long-term scaling.



The Purpose of Creating an MVP



MVP can help you achieve **product-market fit**.

Product-market fit is when your product **successfully meets the needs of a specific target market**, and customers are willing **to buy it, use it, and recommend it.**

Benefit of MVP from Different POV

Business

As a way to achieve a product that users love without having to spend too much money.

Designer

As a means to gather feedback that will later serve as a reference in creating the MVP

User

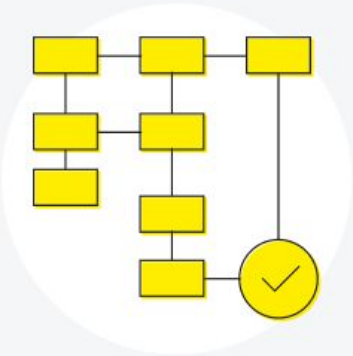
For users, there is no MVP.
They won't realize whether the product is final or still an MVP.

POC vs Prototype vs MVP

The Differences

Proof of Concept vs. Prototype vs. MVP

Validating the idea, including the technical side, and listing the features

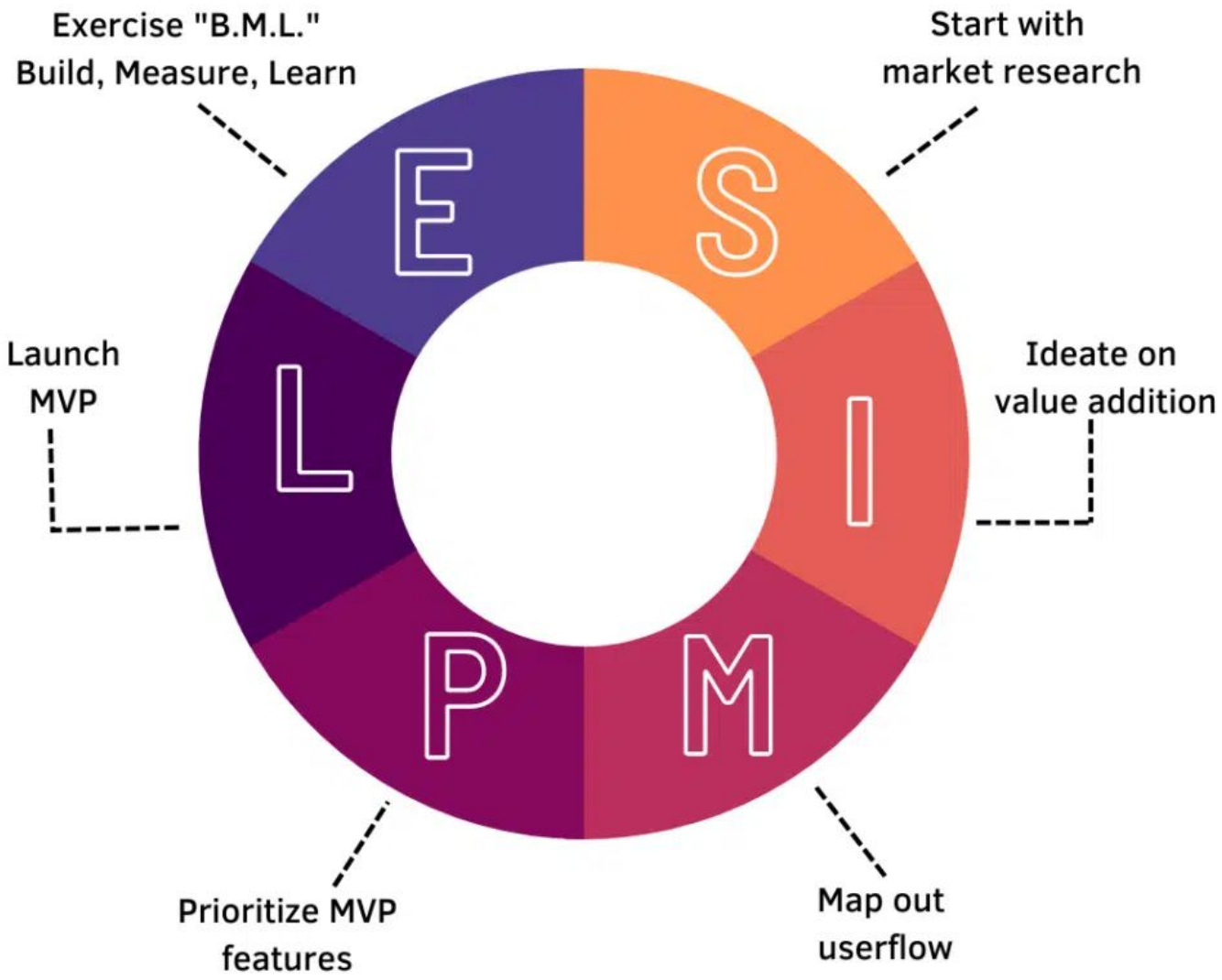


Creating a **product model** to visualize its features, testing it, and improving usability

Building a product's **functioning pilot version** and collecting feedback from real users

Steps to Built MVP

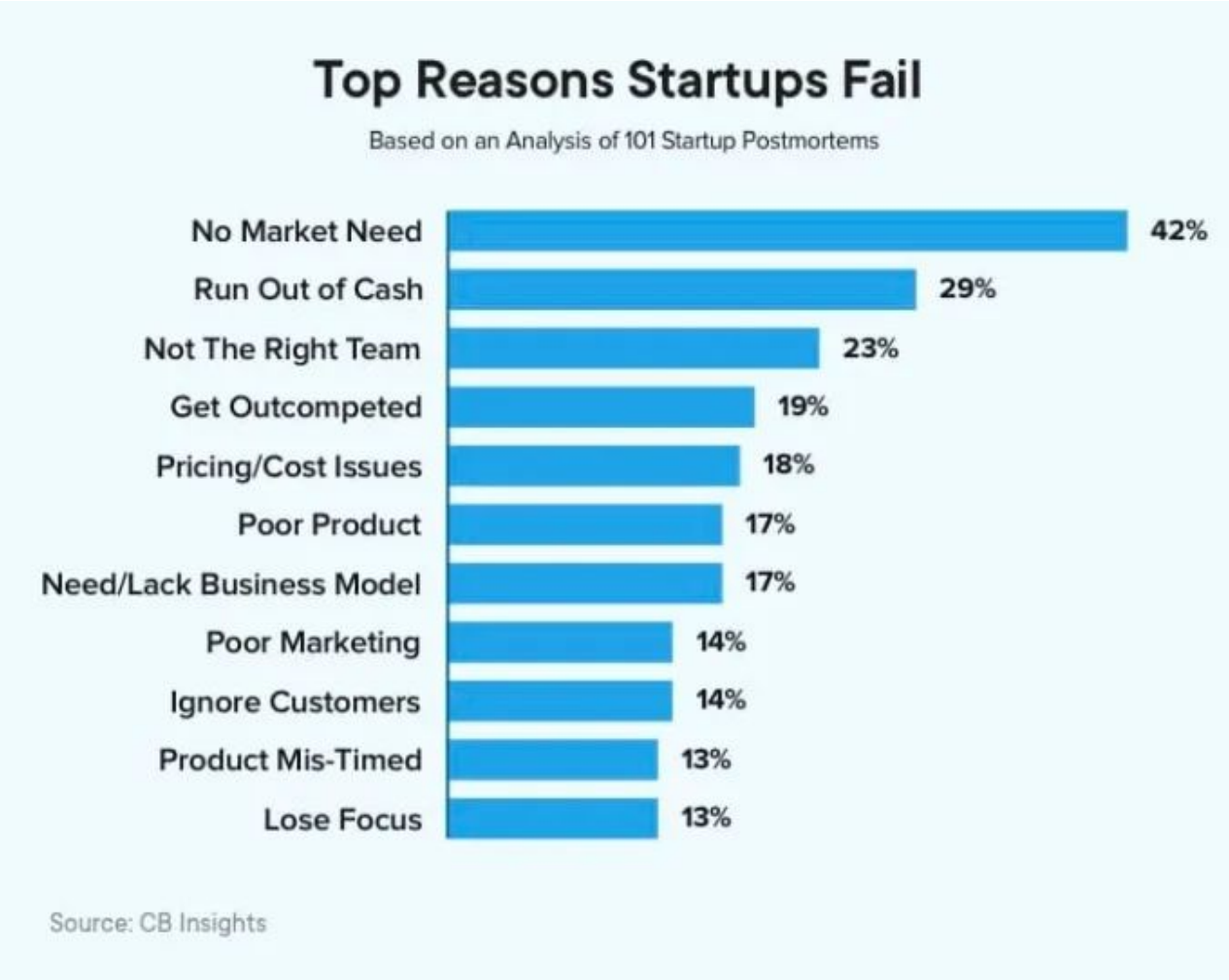
The Framework



Framework **SIMPLE**:

1. Start with market research
2. Ideate on value addition
3. Map out user flow
4. Prioritize MVP features
5. Launch MVP
6. Exercise B.M.L

Start with market research



Ensuring Market Fit Before MVP Development

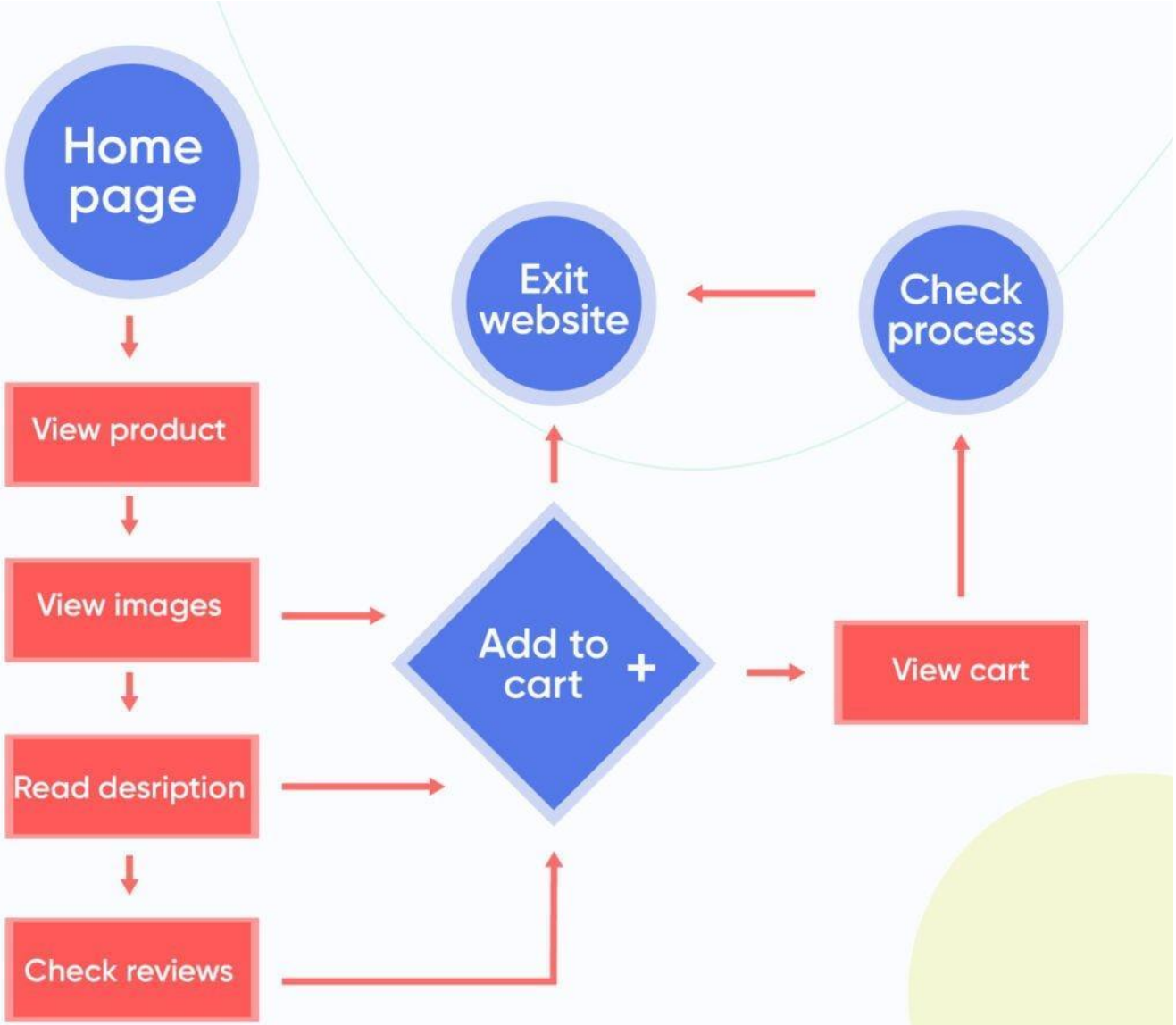
- Validate Market Needs First
  Ensure your idea aligns with the needs of your target users before starting the MVP development.
- Conduct Surveys
  Gather information from potential customers to increase the likelihood of success.
- Monitor Competitors
  Analyze what competitors offer and find ways to make your product stand out.
- Address Market Demand
  Lack of market need is the #1 reason startups fail
- Focus on the Problem-Solution Fit
  If the product doesn’t solve a clear problem, customers will not engage with it.

Ideate on value addition

After conducting research, the next step is to ask the following questions about the product you are going to create:

- **What value can you provide to the user?**
- **How will your product benefit them?**
- **Why should they buy your product?**

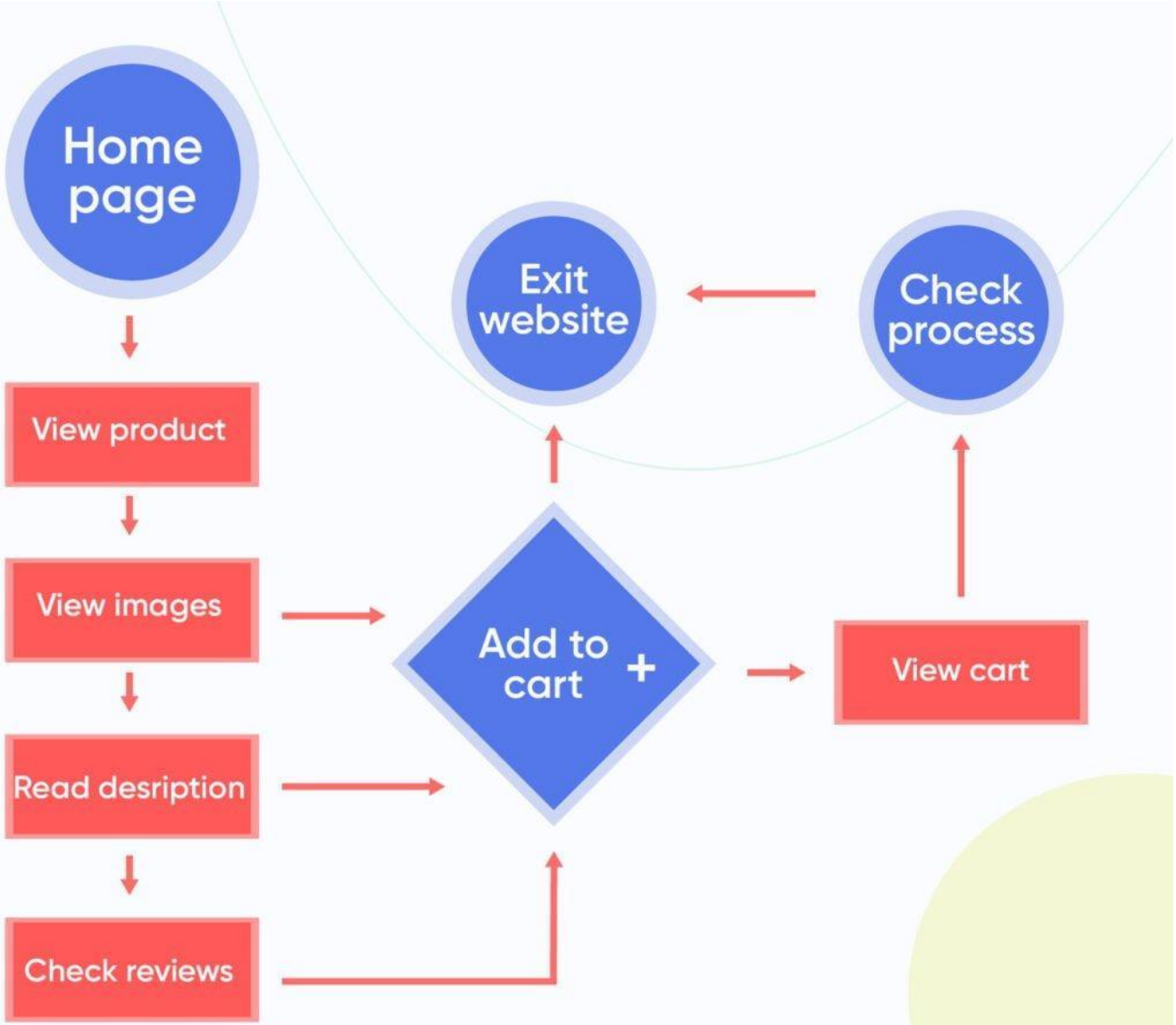
Map out user flow



User-Centered Design
Design the app with the user in mind, from opening the app to completing key actions like making a purchase or delivery.

Focus on User Flow
Ensure a smooth and intuitive flow to avoid missing important steps and prioritize user satisfaction.

Map out user flow



Define Process Stages
Break down the key stages of the user journey to help users achieve their main goals, such as purchasing or managing orders.

Prioritize Basic Tasks
Focus on core tasks (e.g., finding and buying a product) over additional features at the MVP stage.

Feature Definition by Stage
Once the user flow is established, define the specific features needed for each step in the process.

Prioritize MVP features

When creating an MVP, we cannot include too many features.
It is important for us to **prioritize** which features should be included in the MVP.

To prioritize the MVP features, ask questions such as:

- What do the users want?
- Is this product offering them something beneficial?

Next, categorize all the remaining MVP features based on high priority, medium priority, and low priority.
Another essential step is to arrange these features in the product backlog (priority-wise).

Launch MVP

Remember that an MVP does not mean a product with low quality, but your product must meet user needs,
because you will later seek insights on what needs to be changed in the next iteration.

Exercise ‘B.M.L.’ — Build, Measure, Learn

Everything is part of a process: first, define the scope of work, then move the product to the development stage.
After the product development phase, the product must be tested.

After launching the MVP, go over everything again.
The company must get feedback from its clients on the release.
They can determine the acceptance and competitiveness of their goods in the market based on their comments.

Feature Prioritization Method

R.I.C.E Method

REACH	IMPACT	CONFIDENCE	EFFORT
How many people will this feature affect within a given time period? **Example:** customers per quarter, transactions per month	How much will this impact individual users? Use a multiple choice scale: **3** = massive impact **2** = high impact **1** = medium impact **0.5** = low impact **0.25** = minimal impact **Example:** How much will this feature affect conversion rates?	How confident are we about the impact and reach scores? How much data do we have to back up those estimates? Use a % score where: **100%** = high confidence **80%** = medium confidence **50%** = low confidence	How much of a time investment will this initiative require from product, design and development? Measure as persons per month (how much work one team member can do in a month).

R.I.C.E Method

$$\frac{\text{REACH} \times \text{IMPACT} \times \text{CONFIDENCE}}{\text{EFFORT}} = \text{RICE SCORE}$$

After running each feature by this calculation, you'll get a final RICE score that you can then use to rank the order in which you'll build the features. Here's an example:

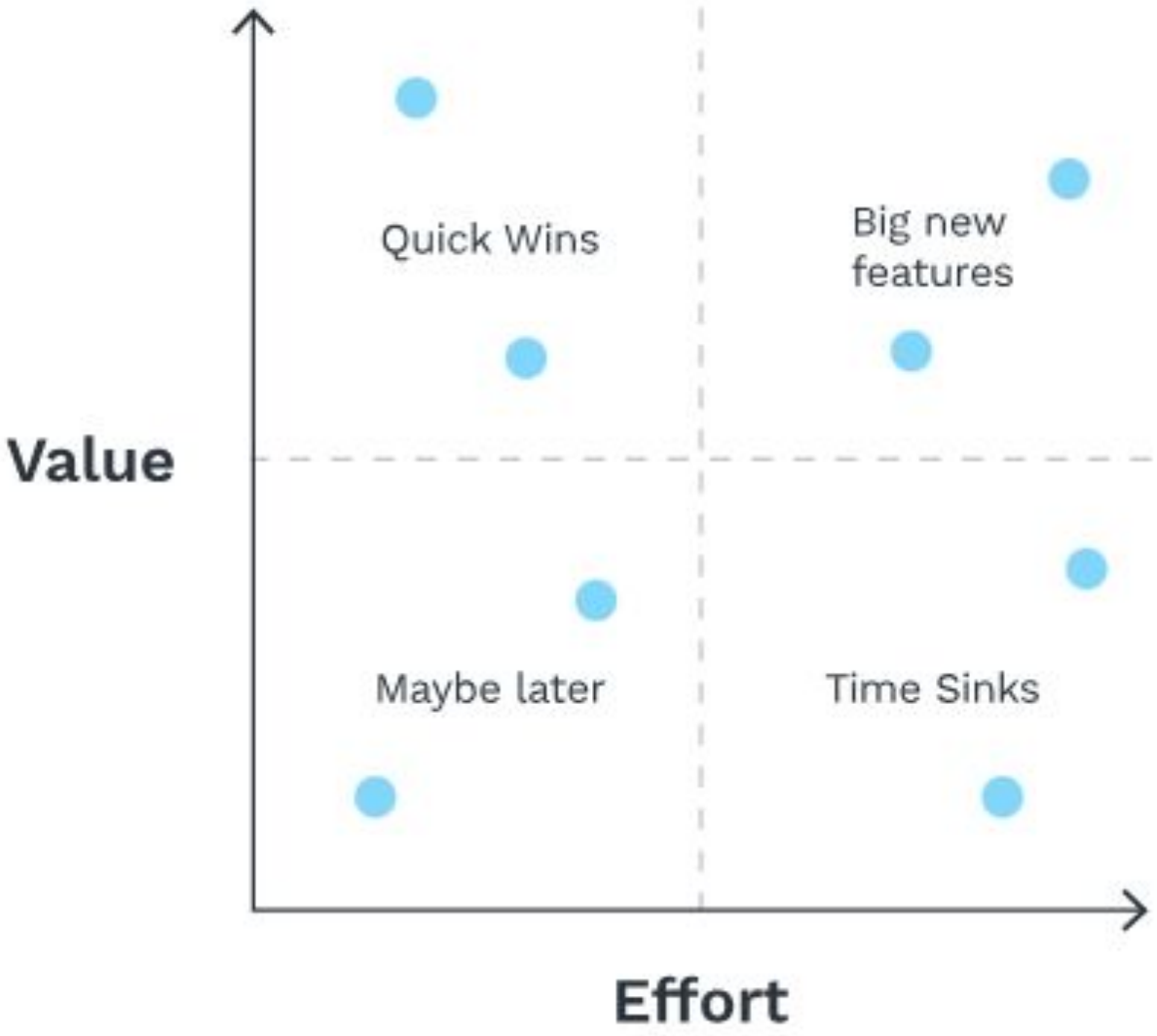
Feature	Reach	Impact	Confidence	Effort	Total
Log in to app with face ID	500	2	80	5	160
Auto suggest transactions	450	2	100	3	300
Budget warning	300	3	80	2	360

Weighted Scoring

	Customer Value	Impact on Business Goals	Implementation Costs	Dev Risks	Total
Weight	40%	40%	10%	10%	100%
Ideas	Scores (out of 5)				Priority Score
Budget warning	3	1	1	2	
Log in to app with face ID	4	2			
Auto suggest transactions	5	3			

	Customer Value	Impact on Business Goals	Implementation Costs	Dev Risks	Total
Weight	40%	40%	10%	10%	100%
Ideas	Scores (out of 5)				Priority Score
Budget warning	3 x 40 = 120	1 x 40 = 40	1 x 10 = 10	2 x 10 = 20	190
Log in to app with face ID	4 x 40 = 160	2 x 40 = 80	5 x 10 = 50	1 x 10 = 10	300
Auto suggest transactions	5 x 40 = 200	3 x 40 = 120	4 x 10 = 40	2 x 10 = 20	380

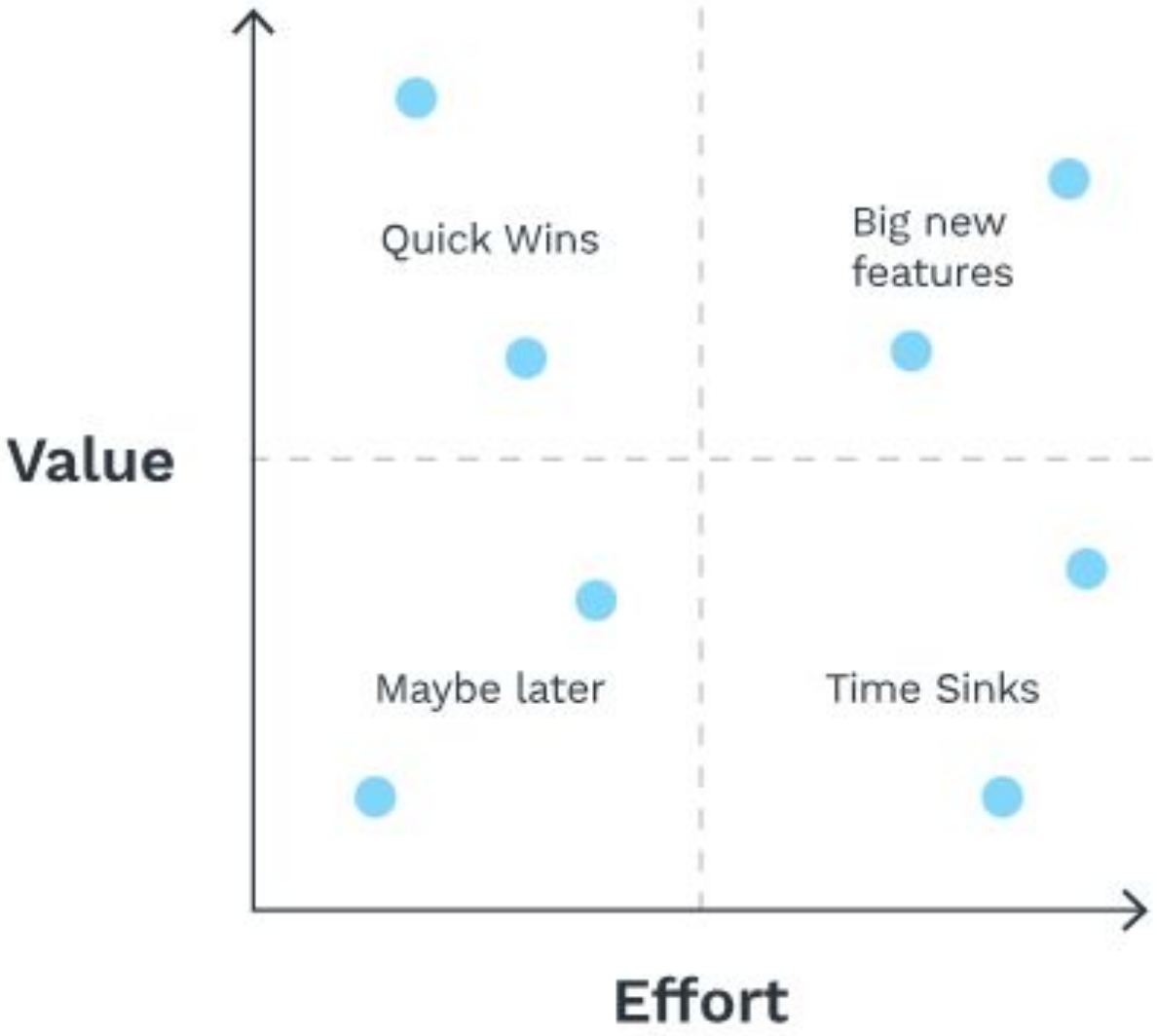
Impact-Effort Matrix



High value, low effort (“Quick wins”): These are the no-brainers, the low-hanging fruit. These are the ideas that don’t require tons of development effort, time, or money. They’re risk-averse, cheap solutions that are relatively easy on the technical side.

Low value, low effort (“Maybe later”): These are the “maybes”, the “will get to it later when the scope opens up a bit more” ideas. They’re not essential to the success of the product, but they’d have a noticeable impact.

Impact-Effort Matrix

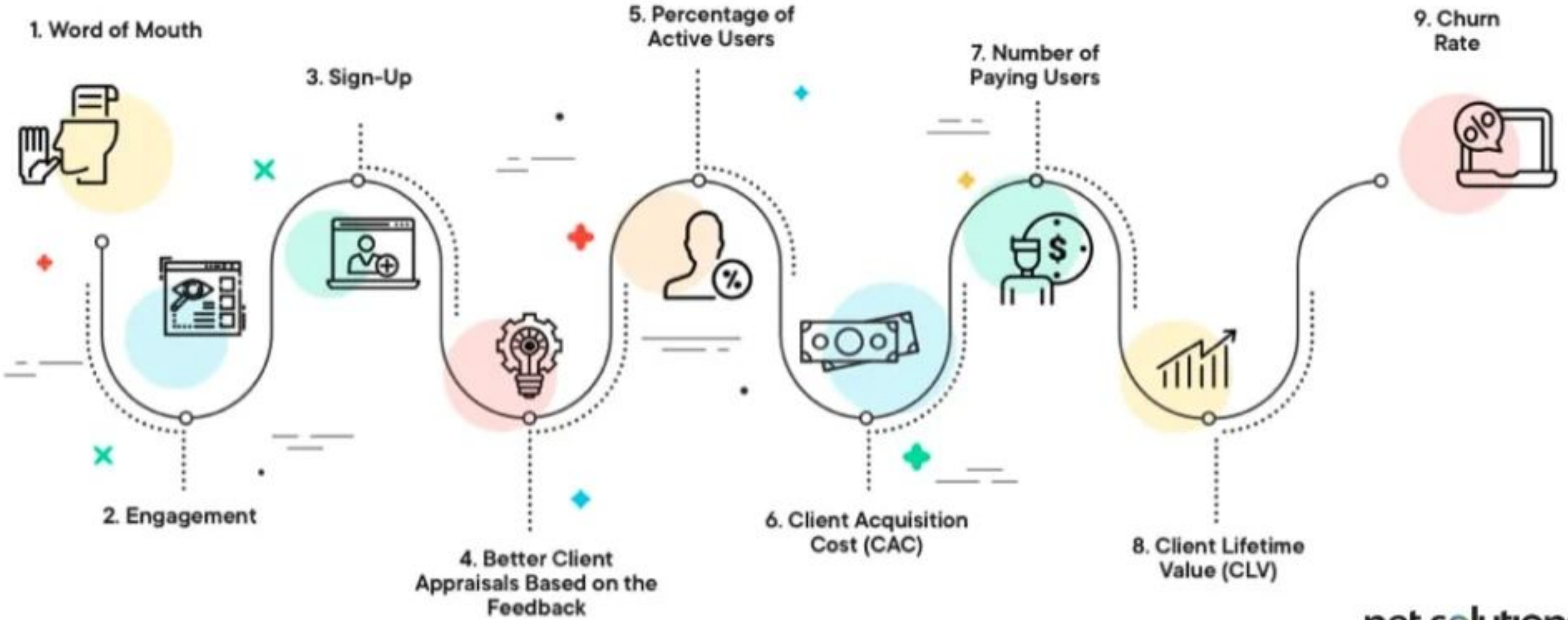


High value, high effort ("Big new features"):
These are the ideas that need a strategic approach. Your team will definitely work on implementing these ideas at some point in the future, they just need to be planned out more carefully.

Low value, high effort ("Time sinks"):
These are the ideas you can afford to pass up. They're not worth doing at the time of the assessment, and they're not likely to become a priority for a while.

Measuring MVP’s Success

Ways to Measure Success After Building an MVP



net solutions

Measuring the Success of MVP

- Engagement
  How engaged are your users with the product you have? This can be seen from how often they access your product in a day and how many features they use.
- Sign-Up
  How many users decide to register to use your product?
- User Feedback
  How do users comment on your product? Is the feedback mostly positive or negative?

Measuring the Success of MVP

- Percentage of Active Users
  How many users are active in using your product compared to the total number of users who have registered?

Active users / total register user * 100%

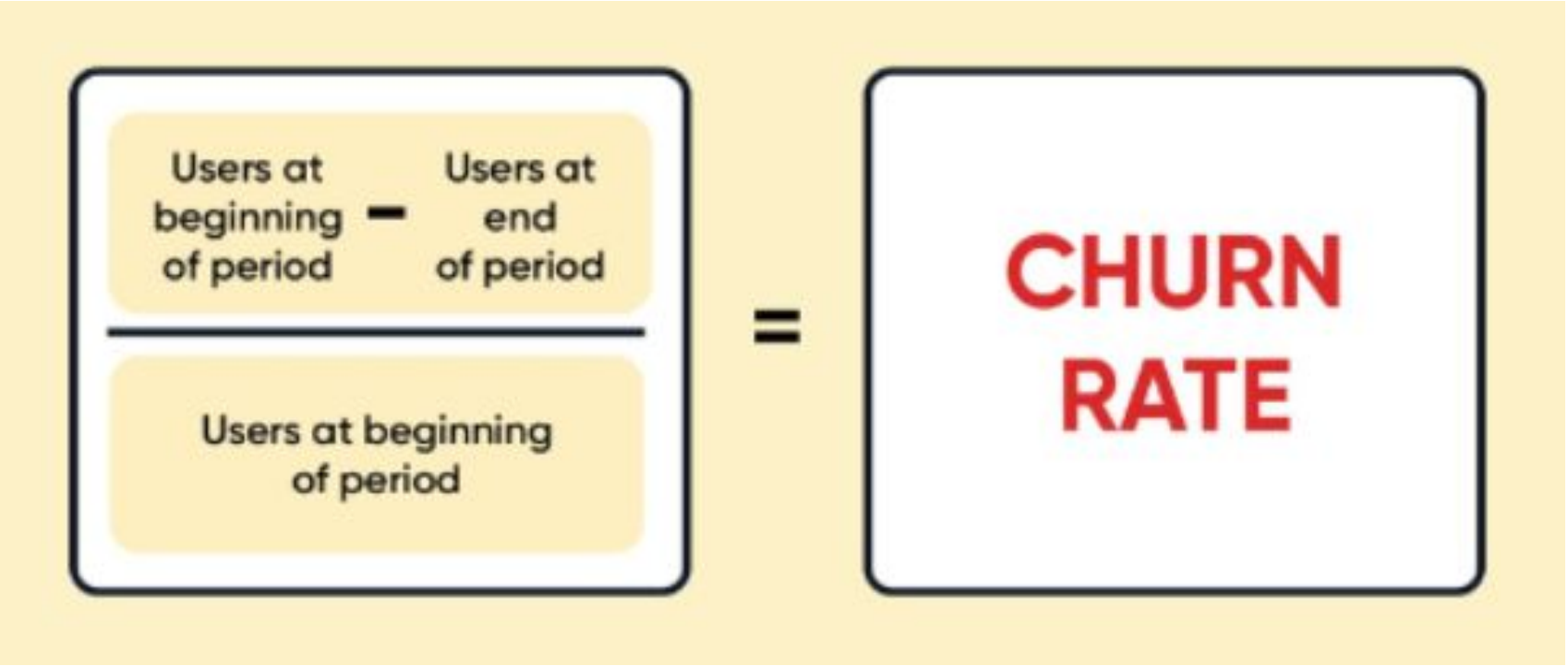
- Sign-Up
  How many users decide to register to use your product?
- User Feedback
  How do users comment on your product? Is the feedback

Measuring the Success of MVP

- Average Revenue Per User (ARPU)
  When creating a product, we definitely expect profit, right?
  So, it's also important to calculate the average spending of users on our product.

Measuring the Success of MVP

- Churn Rate
  Churn rate is calculated to see how many users stop using our product.



MVP Examples

GOJEK

In 2010, Gojek only had 20 motorcycle partners, and users could order a motorcycle via the call center provided by Gojek.

AIRBNB

At the time of its launch, Airbnb rented out apartments owned by the founders for use by users.

Any Question?