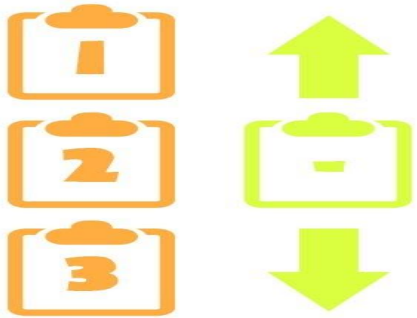
Product Prioritization

By
Uchechi chibuzor
Product Manager

PRODUCT PRIORITIZATION

WHAT IS PRIORITIZATION

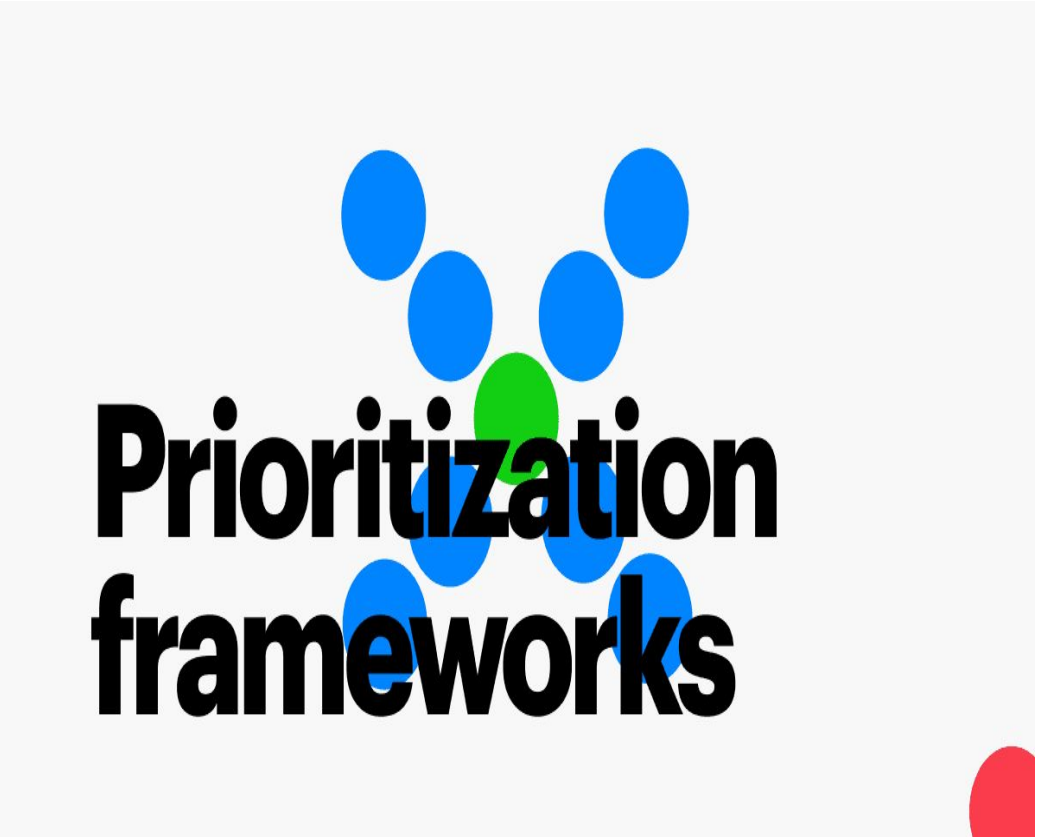


Ranking potential development items in order of importance. In terms of product management, this means determining which themes, initiatives, or features should get slotted into the product roadmap and the next set of upcoming product releases.

Product prioritization isn’t just about making a stack of features in a certain order, it also involves juggling the many inputs and opinions of stakeholders.

Product prioritization framework is a methodology that helps teams weigh their opportunities against various constraints, such as business goals, customer value, product requirements, and available resources.

Some of the most popular prioritization frameworks include:



RICE

RICE allows product teams to work on the initiatives that are most likely to impact any given goal.

This scoring system measures each feature or initiative against four factors: reach, impact, confidence and effort (hence the acronym RICE).

Here’s a breakdown of what each factor stands for and how it should be quantified:

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).

Formula for calculating RICE

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

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

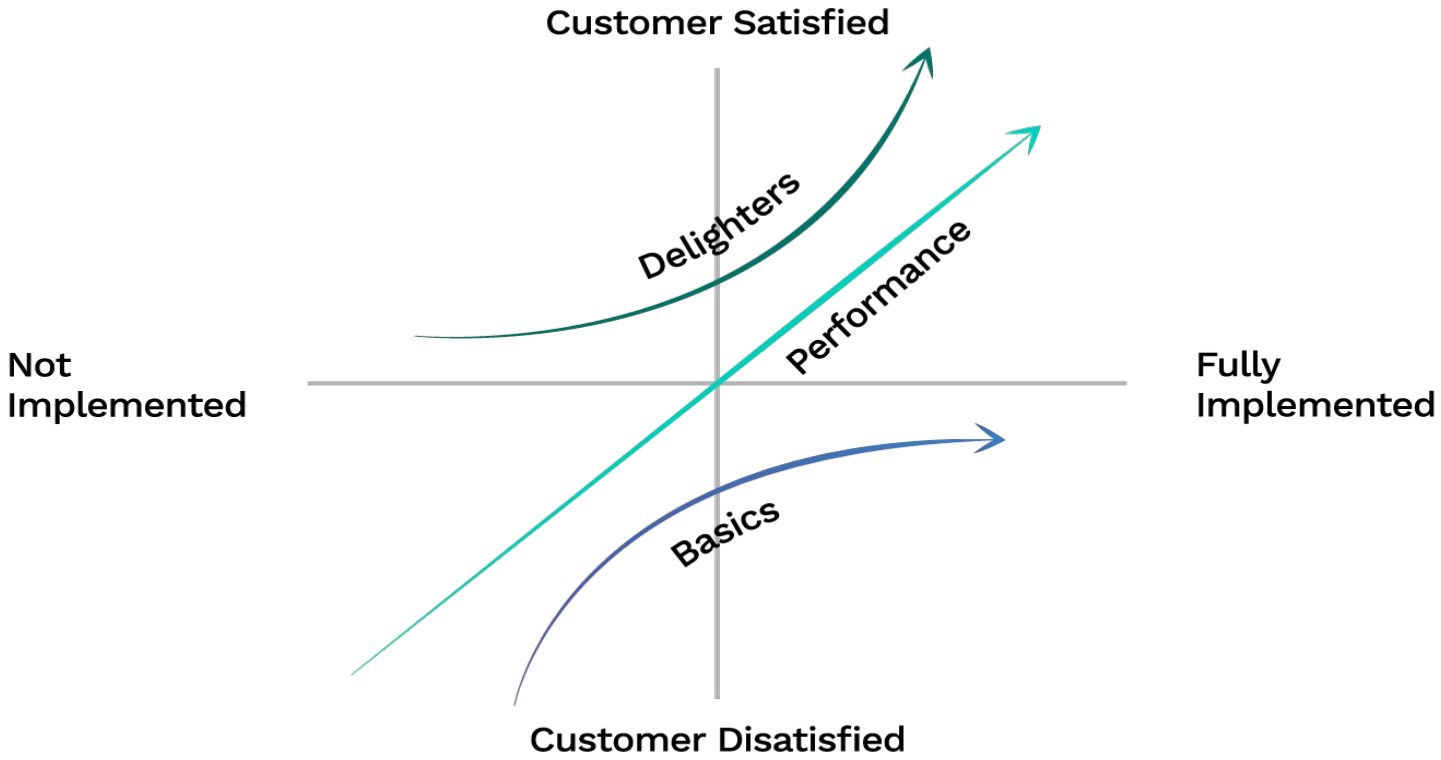
Priority	Idea name	Reach + Positive Factor	Impact + Positive Factor	Confidence + Positive Factor	Effort − Negative Factor	R.I.C.E Score
1	Two Factor Authentication 5 pieces of feedback Auth	72	1 2 3 4 **5**	75%	1 **2** 3 4 5	135
2	Apple Pay Integration 10 pieces of feedback Integration	66	1 2 **3** 4 5	50%	1 2 **3** 4 5	33
3	Reskin Shopping Cart 3 pieces of feedback UI Refresh	48	1 2 3 **4** 5	50%	1 2 3 **4** 5	24
4	Guest Checkout Improvement 7 pieces of feedback UX	46	1 **2** 3 4 5	50%	1 2 **3** 4 5	23

R.I.C.E. Framework

Kano Model

The Kano model plots two sets of parameters along a horizontal and a vertical axis. On the horizontal axis, you have the implementation values (to what degree a customer need is met). These values can be classified into three buckets:

- **Must-haves or basic features:** If you don’t have these features, your customers won’t even consider your product as a solution to their problem.
- **Performance features:** The more you invest in these, the higher the level of customer satisfaction will be. E.g real time tracking
- **Delighters or excitement features:** These features are pleasant surprises that the customers don’t expect, but that once provided, create a delighted response. Eg personalized product recommendation



The Kano Model

The MoSCoW Method

The MoSCoW method allows you to figure out what matters the most to your stakeholders and customers by classifying features into four priority buckets. MoSCoW (no relation to the city—the Os were added to make the acronym more memorable) stands for Must-Have, Should-Have, Could-Have, and Won't-Have features.

- **Must-Have:** These are the features that have to be present for the product to be functional at all. They're non-negotiable and essential. If one of these requirements or features isn't present, the product cannot be launched, thus making it the most time-sensitive of all the buckets.
- **Example:** "Users MUST log in to access their account"

- **Should-Have:** These requirements are important to deliver, but they're not time sensitive.

Example: "Users SHOULD have an option to reset their password"

- **Could-Have:** This is a feature that's neither essential nor important to deliver within a timeframe. They're bonuses that would greatly improve customer satisfaction, but don't have a great impact if they're left out.

Example: "Users COULD save their work directly to the cloud from our app"

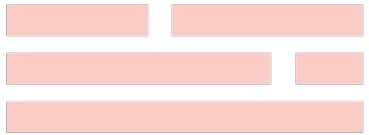
In scope

for this timeframe

Must have

Should have

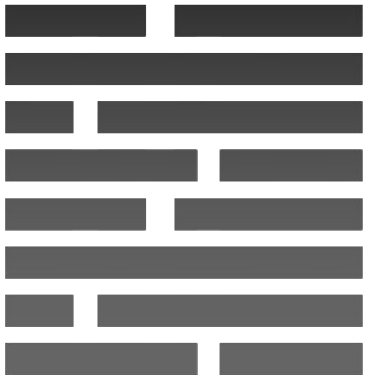
Could have



Out of scope

for this timeframe

Won’t have this time



MoSCow Method

- **Won’t-Have:** These are the least critical features, tasks or requirements (and the first to go when there are resource constraints). These are features that will be considered for future releases.

The MoSCoW model is dynamic and allows room for evolving priorities. So a feature that was considered a “Won’t-Have” can one day become a must-have depending on the type of product.

Value Vs Effort

This simple approach to prioritization involves taking your list of features and initiatives and quantifying them using value and effort scores.

How important is it?

- **Value**
  - Potential revenue
- **Benefit**
  - To current customers
  - To potential customers
- **Impact**
  - On the business goals
  - On the strategic goals

How difficult is it to build?

- **Cost**
- **Effort**
  - Development effort
  - Operational effort
  - Implementation effort
- **Risk**
- **Complexity**

With this method, it’s important to keep in mind that the final scores are just an estimation. A lot of guesswork and opinions (backed with as much applicable data as possible) are involved in the process of quantifying the big question prioritization aims to answer: “Will this feature/update push our goals and metrics forward if we build it? Can we feasibly build it with the resources we have?

This method of prioritization makes room for healthy discussions among stakeholders on what they believe value and effort means, which in turn helps product managers find the strategic alignment holes and fix them.

HOW TO CHOOSE A FRAMEWORK

Few things to consider:

- **MoSCoW:** when you need to include or exclude a feature or a set of features from a release.
- **RICE:** when you need a defined system instead of creating a new one, slot in your numbers and calculate the score.
- **Kano:** when you want to make product improvement or add exciting or wow effects on your users.
- **Value Vs. Effort:** when working on a new product and with scarce resources


Adobe Stock | #51072375