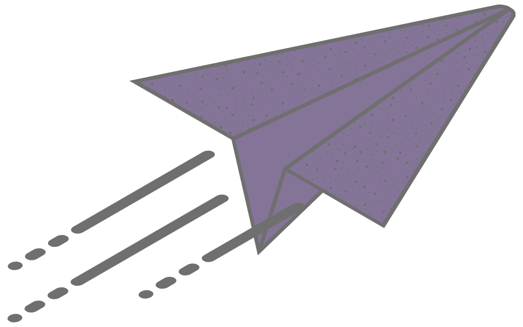


HIGH LEVEL USER-STORIES

Develop high-level user stories to capture the primary functionalities and user interactions.

Document Owner

Product Manager

Responsible Team

Product Management Team

Document Review

- Development Team,
- Stakeholders

Methods & Frameworks

- User research findings,
- user journey maps
- User story workshops
- INVEST framework
- MoSCoW method

Tips & Tics

Focus on the user:

- **Start with user personas:** Define your target audience and create personas representing their needs, goals, and frustrations.
- **Ask "who" and "why":** For each feature, ask who the user is and why this feature is valuable to them. This helps ensure user-centricity.
- **Use user language:** Avoid technical jargon and focus on the language your users would use to describe their needs.

Break down features into actions:

- **Think in terms of tasks:** Imagine the user interacting with the product. What steps would they take to achieve a specific goal?
- **Start simple:** Each user story should focus on a single, achievable goal for the user.
- **Use the INVEST framework:** Stories should be **Independent**, **Negotiable**, **Valuable**, **Estimable**, **Small**, and **Testable**.

Connect stories to features:

- **Map features to user journeys:** Visualize how users flow through the app and which features they use at each stage.
- **Group related stories:** Cluster user stories that address similar needs or functionalities under a larger theme.
- **Prioritize stories:** Use MoSCoW method or other prioritization techniques to focus on the most impactful stories first.

FEATURE & USER STORY

Feature

- A functional element of a software product that delivers a specific benefit to users.
- Features can be large and complex (e.g., a chat system) or small and simple (e.g., a login button).
- They are defined from a technical perspective, focusing on functionality and implementation.

"Mood Tracker"

in a Mental Health Tech App

High Level User-Story

- A brief description of a feature from the user's perspective.
- It describes what the user wants to achieve and how the feature will help them.
- User stories are written in plain language and avoid technical jargon.
- One Feature can have multiple User Stories

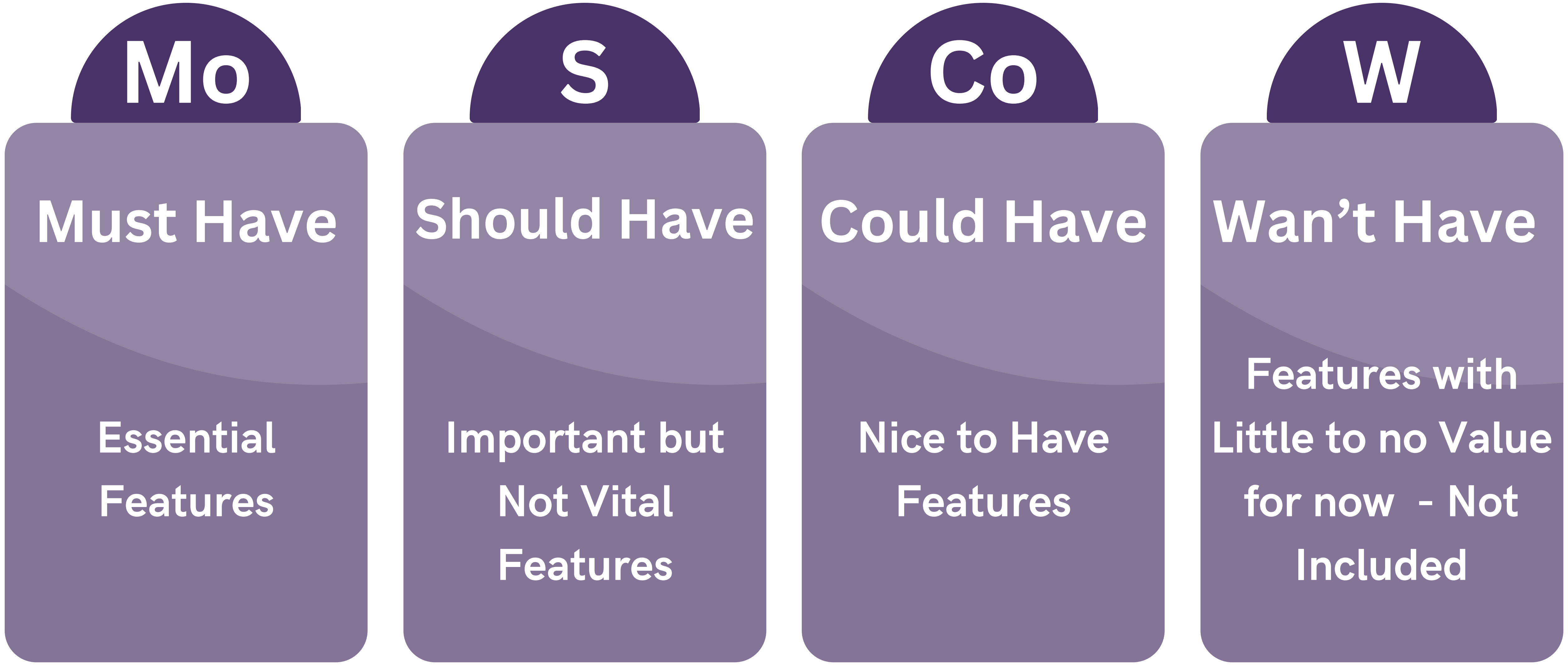
“As a user, I want to add notes to my mood entries to capture specific thoughts or situations that influenced my mood."

MORE EXAMPLES

Feature	High Level User-Story
Mood Tracker	As a user, I want to track my mood daily so I can identify patterns and understand my triggers better. (Focuses on the core functionality of recording mood entries.)
	As a user, I want to add notes to my mood entries to capture specific thoughts or situations that influenced my mood. (Focuses on deeper reflection and context.)
	As a user, I want to see my mood entries displayed in a calendar format to easily identify trends and fluctuations over time. (Focuses on data visualization and insights.)
Guided breathing exercises	As a user who wants to incorporate mindfulness into my daily routine, I want access to a variety of guided breathing exercises so I can explore different techniques and find what works best for me.
	As a user who needs to manage stress in different situations, I want to customize the duration of my breathing exercises depending on the level of stress I'm experiencing.

MoSCoW (Prioritization Framework)

The MoSCoW method plays a crucial role in defining a Minimum Viable Product (MVP) by helping you prioritize features and focus on what's **truly essential for initial validation**.



Mo

Must Have

Essential Features

S

Should Have

Important but Not Vital Features

Co

Could Have

Nice to Have Features

W

Wan’t Have

Features with Little to no Value for now - Not Included

MoSCoW (Prioritization Framework)

Clear Prioritization

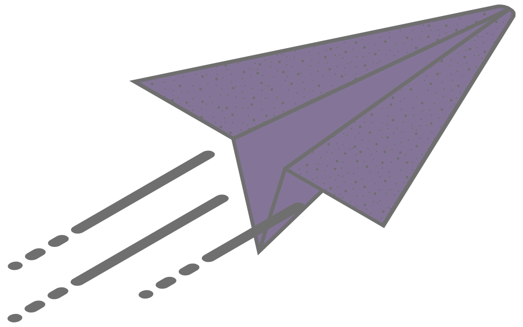
- By categorizing features as Must-have, Should-have, Could-have, and Won't-have, MoSCoW forces you to distinguish critical functionalities from nice-to-haves.
- **Must-haves become the core of your MVP**, ensuring it delivers fundamental value and solves the critical problem you're addressing.

Resource Allocation

- MVP development often comes with constraints like limited time, budget, and resources. MoSCoW helps you **allocate resources towards the most impactful features first**.
- By focusing on Must-haves, you minimize the risk of spreading yourself thin and failing to deliver even the core functionalities.

Scope & Transperancy

- Defining an MVP requires careful scope management. MoSCoW helps you avoid feature creep by keeping the initial scope focused on Must-haves.
- This allows you to **launch quickly and gather user feedback** before investing in additional features.
- MoSCoW provides a **clear framework for communicating with stakeholders**



MINIMUM VIABLE PRODUCT (MVP)

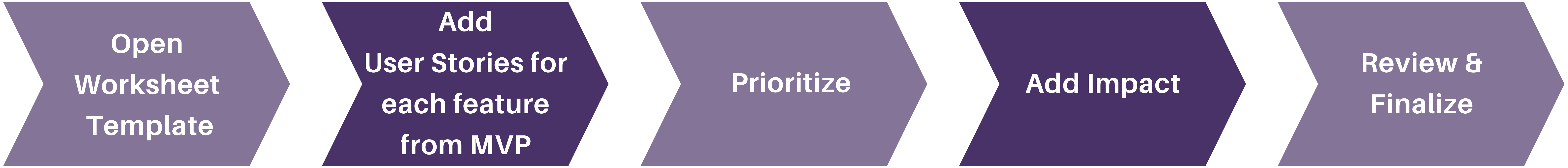
For <<Product Name>>

Replace Logo with your logo or Product Name

productelements

By <<Team Name/Presenter Name>>
<<Date/Year>>

HIGH LEVEL USER-STORIES FOR MVP



Open Worksheet Template

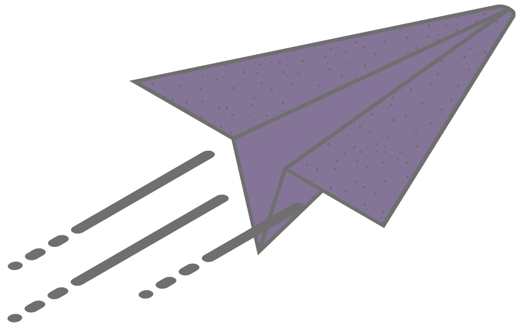
Add User Stories for each feature from MVP

Prioritize

Add Impact

Review & Finalize

Perform all above process on worksheet



THANK YOU

For more such product resources please visit : www.productelements.co

productelements