Product Planning Prompt Pack

maban.co.uk

How to use this pack

This pack is to be used during feature planning, prompting your team to consider things that are often forgotten until it's too late.

It's best to use these as early on in your process as possible. So, next time you start planning a new feature, schedule some time to sit with your team, get out the deck, and go through each card one-by-one.

First, go through them quickly, putting them into 2 piles – one for the items that need to be considered further, and a discard pile for the ones that aren't relevant to your feature. Don't discuss solutions yet, this is just a "yes, relevant"/"no, not relevant" exercise.

Next, pick up the pile of cards that need consideration, and talk through each item as a team. Add notes to your feature plan or tickets to your backlog to cover these and ensure they will be actioned. If there are people you need to consult who aren't in the group, make sure you follow up with them next.

maban.co.uk

SCALE CONSIDERATION

How will this work if the customer wants to do this 100 times?

Can a user can perform repetitive actions in bulk?

MARKET CONSIDERATION

How will we know when this feature is successful?

What does success look like for this feature, and how can it be measured? Bluntly, think about what tracking would be needed to demonstrate that the effort of building this feature was worthwhile.

TECHNICAL CONSIDERATION

How will the engineering team know if this stops working?

Don't rely on customers complaining about broken experiences to learn that a feature has stopped working properly – ensure logging and alerts are in place.

DESIGN CONSIDERATION

Will someone with colorblindness be able to use this?

WCAG 2.0, Guideline 1.4.1: "Color is not used as the only visual means of conveying information, indicating an action, prompting a response, or distinguishing a visual element".

MARKET CONSIDERATION

How is this solved in other products outside our industry?

Don't just look at competitors for inspiration – see how other products are innovating on similar problems.

SCALE CONSIDERATION

What's the performance budget?

What's an acceptable wait time on different devices and internet speeds for this to be ready to interact with? How can this be measured? What alerts will go off if this budget is exceeded?

SECURITY CONSIDERATION

What audit log events should be added?

Write a series of user stories from the perspective of an administrator who is trying to figure out who has performed certain actions related to this feature.

How can a user share this with their team?

If a user wants to tell someone else about this capability, are there any easy ways for them to share it?

How will a keyboard-only user interact with this?

Not everyone has the ability to use a mouse or trackpad. Ensure this is built in a way that is usable with just a keyboard. WCAG 2.0, Guideline 2.1: "Make all functionality available from a keyboard".

What will make this ready enough to be released?

Schedule a Go/No-Go meeting shortly before release to run through what's missing, and make a call on whether to release as-is or continue development.

If this requires a new version to be installed, what will the experience be for legacy users?

What's the best way to inform and encourage users to upgrade? Does the new version have any breaking changes or missing features that users should be notified about? How long will the old version be supported?

What plan will this be available on, and what will the experience be for the users not on that plan?

Consider signposting this feature for users who are not on the plan, so they are aware that it exists, encouraging them to upgrade.

How does a user undo this if they make a mistake?

Can a user revert or roll back changes, or make edits and delete data?

What, if any, prerequisites are needed before someone can use this?

Does the user have to install any software, or upgrade the version they're on? Do any feature flags or entitlements need to be applied?

How can this be rolled back if there are problems?

Mistakes happen. Make rollbacks less painful by using feature flags, and ensuring the release commit is isolated so it's easy to revert.

Where can someone in customer success find out more about this?

When a new customer success manager joins, where can they find additional information such as the environment this works/doesn't work in, and what additional capabilities are still planned or in development?

SECURITY CONSIDERATION

What personal user data could be exposed?

Think carefully about whether any personal data is being transferred or logged, and if it is, ensure it's compliant with GDPR.

MARKET CONSIDERATION

Where can someone find out how to use this?

Ensure there's external documentation, and ideally link to it straight from the feature itself.

SCALE CONSIDERATION

What if this feature is really popular and gets 100 times the expected usage?

What needs to be put in place to ensure that the high levels of usage don't cause a negative impact on the rest of the system? Can alerts be put in place to give an early warning? Could performance-intensive capabilities be easily disabled?

SECURITY CONSIDERATION

How could someone abuse or exploit this?

What could someone do if they wanted to use this feature in a malicious or unethical way?

TECHNICAL CONSIDERATION

How will someone interact with this via the API?

Even if the feature doesn't have an API, consider how one would work anyway to help the team focus on key actions and use cases.

MARKET CONSIDERATION

How is the competition doing this?

Does the competition have this capability? If so, how can this be made better that what they're offering? Ensure the team is well versed in this feature's market differentiation.

DESIGN CONSIDERATION

How will this work on a mobile/small screen device?

How will the UI adapt to small screens? And what will the experience be like for touch?

TECHNICAL CONSIDERATION

What errors will be needed?

Consider success and failure states, as well as internal error logs. Ensure the messaging is written clearly and helps the user or on call responder understand the error.

SCALE CONSIDERATION

How will this scale if 100 times the expected amount of data or content is added to it?

Consider the extremes, and ensure the feature's design and performance won't be negatively impacted by someone using this extensively.

Which customers should be asked to try this out first?

Gather different types of customers to get a broader range of perspectives – from role size to industry to environment.

How will this work for offline users?

What needs to be done to ensure this works for users in an offline environment, using an on-premises installation, or whose connection has briefly dropped during usage.

Will a screen-reader or search engine be able to read this?

WCAG 2.0, Guideline 1.3.1: "Information, structure, and relationships conveyed through presentation can be programmatically determined or available in text".

How will someone find out this exists?

Don't rely on documentation alone as a way of users finding out about the feature. Ensure it's signposted in places users will see.

## What are the risks of releasing this?

Think of some worst case scenarios on releasing this feature, and what can be done to reduce or eliminate them.

What will a press release say about this?

Writing a mock press release before starting work on the feature to share internally, and to set expectations on what the initial release could look like.

How will this look if a user prints it?
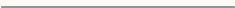
Adding a print stylesheet will help ensure that important information is visible and readable if a user tries to print the page.

The rest of these cards are blank so you can add your own prompts.