

Product Management - Project Preparation & Getting Started

Gavin Munro

1. What is an epic?
 - a. The purpose of an epic is to describe the end-goal job to be done. Epics provide direction for the feature during production. Refer to the epic template after these .
2. How to break an epic into logical user stories?
 - a. A user story is the 'why' of a task. We need to understand why we are adding a feature or making a change, without the 'why', we don't have direction and we can't create valuable products.
 - b. User stories help us gather direction on what needs to be accomplished so we can provide the most value for our customers. They are vital because they show that we [product] understand the pain points users are facing, which ultimately helps us deliver valuable products.
 - c. A user story has two key parts, the context and the desired result. We then know why we are making changes, and what it will accomplish for the user.
 - d. **User Story Template:** As <persona(s)> I need to be able to <use case>, but experience <pain point>. Beach: <persona(s)> are able to <use case> by <solution>, achieving <result>.
3. How to partner with UX and engineering
 - a. During the discovery process you will reach a point where user feedback is building and you have identified key components that need to be included in the feature. Working with UX will help you visualize these features/components and present them to users so they can continue to provide feedback.
 - b. Product determines what gets built, engineering determines how to build it. Once discovery is complete and the mockup is prepared from UX, we go into our task management software and break up the epic into manageable tasks. Each task has a user story explaining the value being delivered by the task, design requirements if applicable, and task requirements. Once these are outlined we provide a summary of what the task should accomplish by detailing Acceptance Criteria. This provides all the information engineering needs to build, design, and test the feature/task.
4. How to perform UAT
 - a. User Acceptance Testing is a key piece to the success of a feature. Once engineering has built the feature and QA has performed testing, we notify specific users (typically our go to super users) and allow them to go into the application to test the new feature or changes. The purpose here is to get their final approval so they have the proper expectations

when it gets released to production, and we can catch any final bugs if they exist.

Epic Template

The following template was created by: Bindiya Thakkar

See original post at:

<https://productcoalition.com/how-to-write-epics-and-user-stories-best-practice-1de5b983900>

INTRODUCTION

In the introduction, you should include about 'why' and 'what' about the epic. If you are writing an epic to develop a new feature include why you have decided to develop this feature, what are the user needs you are trying to solve and what are you trying to achieve with this new feature along with the metric you plan to use for measuring the achievement of the new feature. In addition, if you have any documentation or early wireframes for the feature include those in the introduction to provide as much clear picture & information you can provide to your team. Note: shared common understanding and goal is a key to successful delivery.

In short, your introduction can include:

- summary of what the features you're building are for and why you're building them
- what metrics you are trying to improve
- links to specific documentation
- marketing plans, legal requirements (if any)
- early wireframes

Example: Suppose you and your team is working with a messenger service and now you want to add a feature for the user so that the user can upload the pictures from their device and share it with their contacts. So this is how your introduction can look like:

Introduction

This feature will allow users to send pictures to their contact through our messenger service.

Our research shows that 60% of users would use this functionality and in user testing our designs were intuitive for users.

We expect that this functionality will increase the number of messages sent per user by 10% on avg, which will result in approx. 3% additional time spent using our service in the course of a month.

Ideally, we want people to be able to share multiple varieties of files, but for MVP we have decided only allow uploading photos from their mobile, desktop and tablet.

PRODUCT REQUIREMENTS

An essential part of the epic where you provide with an explanation for the whole team working on it to understand what are they going to design, build, test or release. For example, if you are building a feature that the feature has to be fast or should be available in multiple languages, or should work on multiple devices like mobile, tablet and desktop should be mentioned in the product requirement section of your epic.

Continuing the above example your product requirement could look like below
Product Requirements

- User can initiate a photo message from the message window
- User can select a photo from their own device
- User can preview the image before sending
- User can cancel the send before it is complete
- User should be able to send any resolution photos from their device
- User should be notified about the successful upload
- User can see the uploaded photo in a preview mode and can click to open a full image

Out of Scope:

- User can initiate a photo message from the message window
- User can select a photo from their own device
- User can preview the image before sending

DESIGN REQUIREMENTS

While writing the design requirement collaborate with your UX designer as much as you can. Take their input as there might be things that a designer thinks is important in order to have a better user experience which wouldn't cross your mind. For example, a designer might think the preview should be of a certain size and the profile picture should always maintain certain resolution in order for a good experience than those kinds of requirement should be written here.

Continuing the example ahead this can be how your design requirements may look:
Design Requirements

- The photos should be stored in our servers so the user can see them even when they switch their devices

- Maximum photo size should be 2000X2000 pixels.
- Preview should be atleast 600X600 in the aspect ratio of the uploaded image
- Loading indicator to be shown to the user while the user is waiting for the upload, in case of any delays
- Success indicator to be shown for successful uploads

ENGINEERING REQUIREMENTS

Similar to the design requirement in this part of the epic try to involve the engineers or tech lead as much as possible. Their inputs in the early stage will be very useful while estimation and building it correctly. For example, the engineering team might want to build an API to integrate with some other system in order to fetch and maintain the quality of an image, those kinds of specifications and requirements should be mentioned under engineering requirements.

Continuing with the above example the engineering requirement can look like below:

Engineering Requirements

- Need a database that can scale to X number of images at a maximum of 5MB per image
- Pull user IDs from user profile to connect with the photos in our database
- Create an auto-delete system to delete images after 1 year to make it GDPR compliant

Having a good epic spec document will have a very positive effect on your team to collaborate and build the right thing. For you as a product manager to create user stories to slice it down and prioritize your backlog, have a smoother development process and easy shipment.

USER STORIES

User stories are basically the break down of an epic in a more user-focused way for the engineering team to understand the product requirement. In agile methodologies, everything that we build should be focused around users and hence the main purpose of the user story should be to shift the focus around a feature in a more human conversation manner.

Here is a simple template that is widely used while creating user stories:

As a (type of user), I want (some goal), so that (reason).

The point of the user story is to clearly state the feature desired from the point of view of the user.

A user story must have just the right level of detail. It should be a high-level requirement with additional detail added to the accustomed acceptance criteria. The

acceptance criteria are the clear picture for the engineering team to understand 'what' they are building and for the QA to clearly state the acceptance test. The components to be included are:

- User story
- Acceptance Criteria
- Design attached to the user story

As we have already created an epic, to build a feature that allows the user to share the photos from their devices with their contacts. I would like to continue with the same example to write some user stories.

User story: As a user, I want to select photos from my device in the messenger service so that I can share it with my contacts.

Acceptance criteria: Given I am a user and I click the "add picture" button in the message window, I am presented with a popup window to choose the file I can upload, submit it with the upload button and see a preview of the uploaded image.

Given I am a user who has successfully uploaded a photo from my mobile when I click send, the image is sent to my contact and it appears in our chat window.

If you want to learn in-depth about how to write acceptance criteria you can read the article on [How to write acceptance criteria? Best practices, format, examples, and tips.](#)