http://www.romanpichler.com/blog/user-stories/epics-and-ready-stories/

Epics and Ready Stories

Posted on Wednesday 7th November 2012

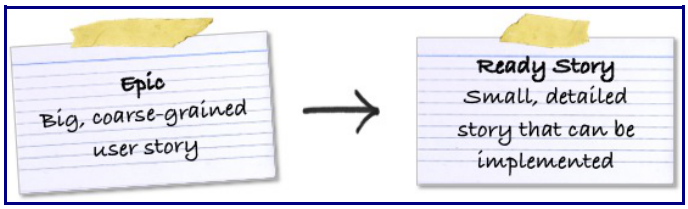
Summary

This post explains how to write user stories at the right level of detail, and how to derive small, ready stories from big, coarse-grained epics.

One of the things I love about user stories is their flexibility. A user story can be big, medium-sized, or small. This allows us to sketch a product with big stories, and then progressively add more detail and refine them into smaller user stories, as we learn more about how to meet the user needs.

But this flexibility comes at a price: I often see user stories that are too small or too big, that contain too much or not enough information. Deriving the right stories, and getting the level of detail right can be challenging.

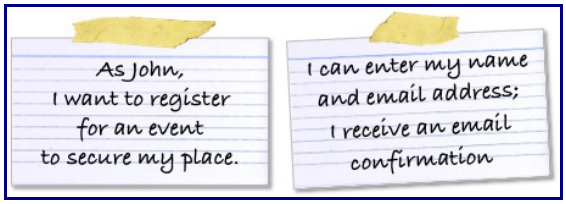
This post suggests a simple, yet effective solution: distinguishing between two types of user stories, *epics* and *ready stories*, and deriving the ready stories with the help of a shared sprint goal or hypothesis.



Epics

Epics are big, coarse-grained user stories. An epic sketches a feature or bigger piece of functionality. It acts as a placeholder for more detailed stories. Think of the epos Odyssey. It's a collection of stories about the adventures of Ulysses including meeting the Sirens, and defeating a Cyclops.

Epics allow you to sketch the product functionality without committing to the details. This is particularly helpful for new products or major product updates, as it buys you time to learn more about the users and how to best meet their needs. It also reduces the time and effort required to integrate new insights: If you have lots of detailed stories, then it's often tricky to relate the feedback you receive to the right stories, and it can be a big effort to change them without introducing inconsistencies. Let's have a look at a sample epic:

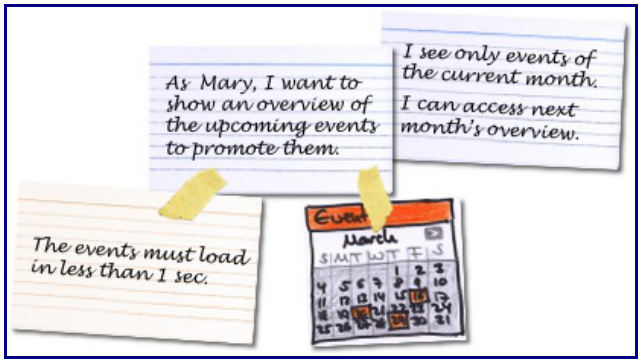


The epic above tells us that the persona John wants to register for an event. The details of the event and the registration are left open for now. Similarly, the acceptance criteria - captured on the right card - are sketchy too. The details will emerge, as we learn more about the event registration by developing software and exposing it to the right people.

Ready Stories

Ready stories are small, detailed stories that can be implemented. These stories have to be clear, feasible, and testable: Everyone should have a shared understanding of the story's meaning; the story should not too big or complex; and there has to be an effective way to determine if the functionality works as expected.

A ready story should also take into account additional aspects: the user interface design and the operational qualities that are specific to the story. Examples of the latter are performance or interoperability. I prefer to work with a paper-based design sketch to capture the user interface, and constraint story cards to describe the qualities, as the following picture shows.



The ready story above is much more specific and detailed than the sample epic discussed earlier: The details enable the development team to implement and test the story successfully.

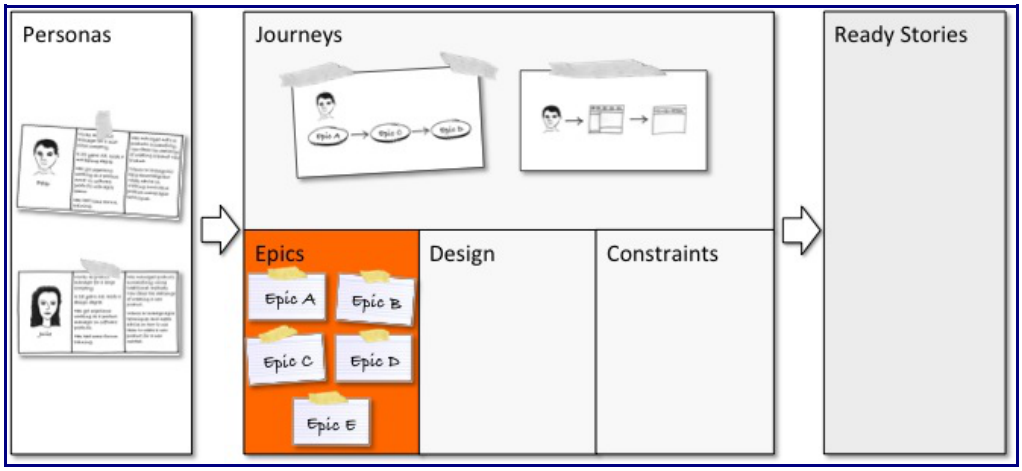
From Epics to Ready Stories

So far we've simplified things by distinguishing between epics and ready stories. But we haven't discussed yet how ready stories are derived from epics. My solution is to identify a sprint goal or hypothesis for the next iteration before the ready stories are written, as the following picture illustrates:

Step 1: Write the Epics

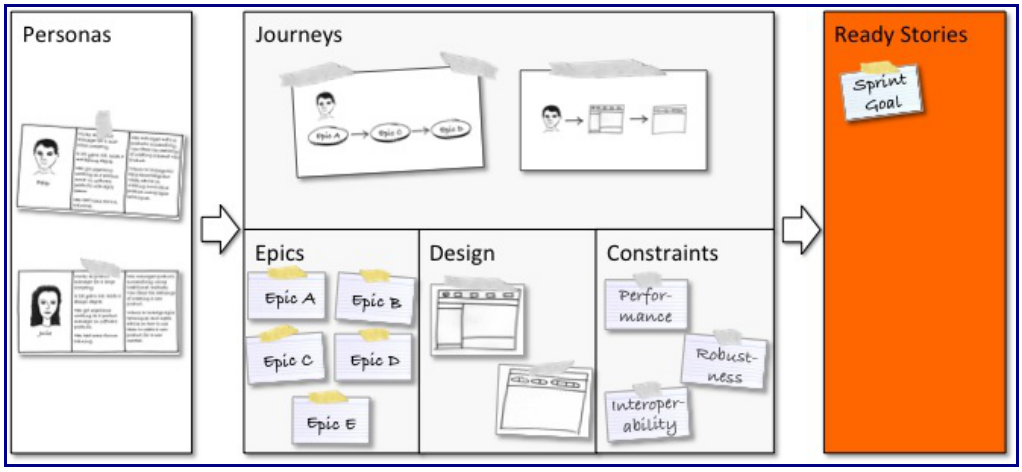
Let me show you how this process can be applied using my Product Canvas tool. The canvas is essentially a multi-dimensional backlog that allows you to describe your product holistically.

The Product Canvas supports working with epics and ready stories by providing dictated sections. It also offers the context for finding the right epics: The user journeys - scenarios, workflows, or storyboards - are a great hunting ground for epics. The epics are placed on the "Epics" section, as the following picture illustrates.



Step 2: Select the Goal of the Next Sprint

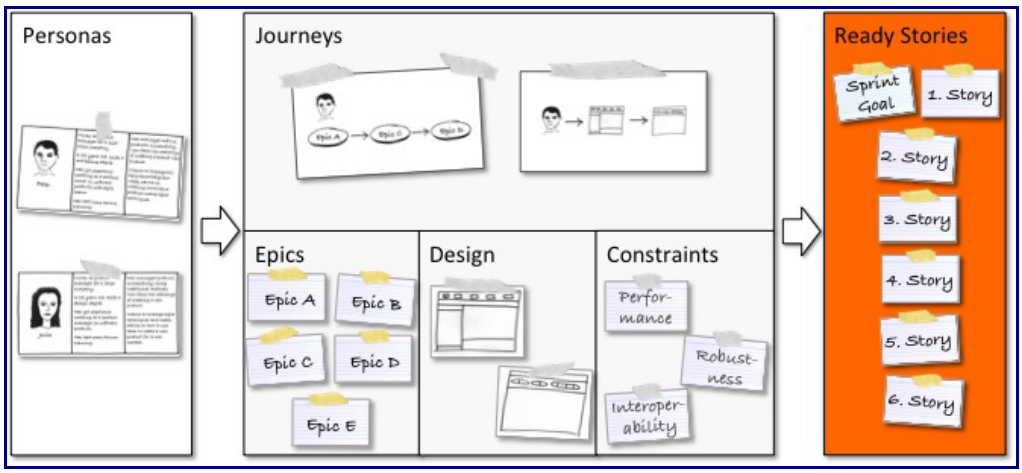
Once the epics sketching the product's main functionality have been written, I populate the design and constraints sections on the canvas. Then I select the goal of the next sprint - either to address the most critical risk and/or to deliver functionality to the stakeholders including the users. Examples of a sprint goal are: "Validate our central user interface design ideas", or "Be able to release epic B". The sprint goal is placed at the top of the Ready Stories section, as the following picture shows.



I find it very beneficial to involve the entire team, the product owner and the development team, in selecting and formulating the sprint goal. This leverages the team's collective knowledge, and ensures that the goal is shared. The latter ensures that the entire team is moving towards one goal. This encourages teamwork, and it makes it easier to analyse the feedback.

Step 3: Derive the Ready Stories

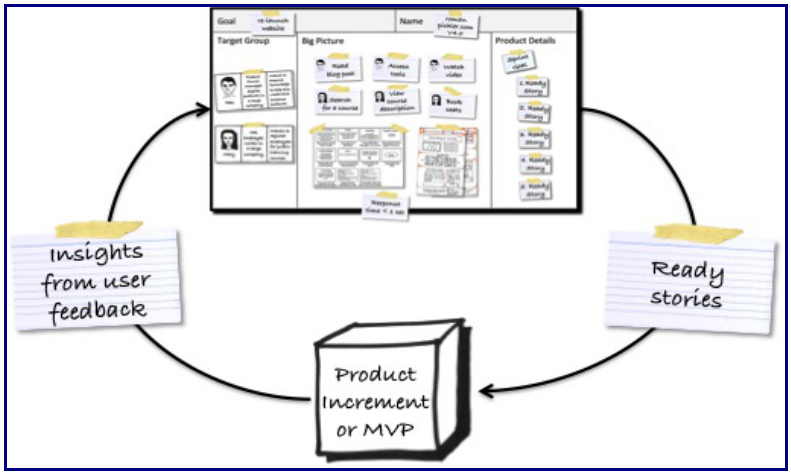
After selecting the sprint goal, I determine which epics contribute to the sprint goal, for instance, epic A and C on the canvas above. I then derive small stories from the appropriate epics, and I order the new stories depending on their importance to reach the sprint goal, as the picture below illustrates.



Finally, I ensure that each story is ready and has the necessary sketches and constraints attached. The entire teams should carry out this step to ensure that the stories are clear, feasible, and testable. The right amount of ready stories are then pulled into the sprint and implemented.

Step 4: Update the Canvas and Repeat the Process

After exposing the outcome of the iteration to the right people, the insights gained from analysing the feedback are worked into the canvas: Existing epics are adjusted or removed, new epics may be added. Then the next cycle starts. A new goal is selected, and new ready stories are written.



Summary

Working with epics helps you sketch the product's functionality, and ready stories provide the input for the next cycle. This focuses your user story writing effort when developing a new product or new features. Employing a shared goal or hypothesis makes it easier to decide which ready stories should be written, facilitates teamwork, and makes it easier to analyse the feedback.