Template for the User Story with full details

Intro: User Story as Library Card

Rarely development teams are working on User Stories containing only in 3 sentences: “As a... I want... So that...”

In most cases User Stories need mock-ups, Acceptance Criteria, design documents, problem statements and/or other details to be discovered and documented.

Jeff Patton (the author of the book about User Story mapping) suggests that a User Story is like "a card in a library card catalog":

> "If I pick up a card from a card catalog, it’s going to have just enough information for me to identify the book. It likely has a title, the author name, a description, the page count of the book, the category of the book and a code to a location where I can actually find a copy of the book in the library.
>
> The card is just a token that is easy to find and organize. No one confuses the cards with a book.
>
> The card catalog is handy because it takes a lot less space than thousands of books would. And I can organize cards in different ways — by author, for example, or by subject.
>
> Your stories will work the same way; that is, you may write them on cards, keep them in a list in a spreadsheet, or enter them into your favorite tracking tool. In a library, you know there’s a book out there somewhere, and if you have identified the right card filed away in the card catalog, it’s easy to find it."

In the library the card ultimately leads to a book with dozens or hundreds pages of information. The User Story in the same way leads us to the details about the functionality that needs to be implemented in order for this User Story to be completed.

This detailed information can be researched and documented before/after/as a result of Backlog Grooming meeting.

User Story with full details

The goal for adding full set of details to User Story is to provide as much information as possible for development Team. It can be done in dedicated confluence page, in Jira ticket using standard and specially defined fields, or in other tools which Product Owner and Team find convenient.

The idea is for Development Team not to lose time during the Iteration or Iteration Planning waiting for answers and go into development with less of assumptions and more of externalized and clear descriptions approved by Product Owner.

Of course, no details page will answer all potential question before actual development begins. The goal is to tackle those that **actually can** be identified and answered before the start of the Iteration.

Structure of User Story Details

The following is the list of types of information that the User Story Details can contain. This list is in no way exhaustive.

User Story text

For the purposes of providing the context for the reader, the text of the User Story ("I as ... Want to ... So that ...") should be included here even if the Master copy of the User Story is kept in Master copy of Product Backlog in separate location.

Detailed description

Here more Story-specific context, specific scope limitations and additional information can be provided as well as a list of Open Questions to be addressed.

Problem statement and value description

These types of context information are not critical for development per se, but they might provide unexpected emerging value: Development Team understanding the business problem might come up with useful insights / suggestions / innovative solutions that are not obvious to other stakeholders. Also knowing the details of the business context helps with accurate modeling and system design on all levels.

Formal Workflows

Simple flowcharts and sometimes even complex diagrams (e.g. UML Activity or Sequence Diagrams, etc) in case of non-trivial business workflows.

UI layout

It is very useful to have a clear layout of the UI before Iteration Planning meeting - it helps adequately estimate complexity, break down tasks and even find better UX suggestions as a result of Team discussions.

So any subsets of the following UI-artifacts will be helpful:

- UI sketches (e.g. photographs of UI sketches from whiteboard)
- Wireframes
- Mock-ups
- Screen-by-screen specifications or Storyboards, etc.

Data models /mappings / examples

The data used in the workflow of particular User Story should be researched and formalized:

- types of data-sets, documents and similar data-related artifacts,
- description of their formats,
- mapping of data fields to the business terms and sometimes to the XML/JSON/DB fields/property files,
- expected data models / database schemas / ER-diagrams,
- etc.

Constraints / Non-functional requirements

Whatever the way that Product Owner and the Team decided to document non-functional requirements and constraints for the Product, the links and mentions of all those applicable to specific User Story need to be included in its detailed description.

Preconditions

Some defined conditions which need to be true in the System for the User Story to work and to make sense.

Acceptance Criteria / Acceptance Tests

It is crucially important for Product Owner to create or (as a minimum) to sign off on the list of Acceptance Criteria for each User Story. Using these criteria completed User Story will be presented during the Sprint Review Meeting.

Broadly accepted format for the Acceptance Criteria is the "Given/When/Then":

Given	**Given** some precondition
When	**When** Persona does some action
Then	**Then** they obtain defined result

Some teams make distinction between Acceptance Criteria and Acceptance Tests, making the former more broad description of the User Story successful implementation and the latter - more formal tests. The important thing here is to have a consistent and agreed upon approach and to follow it.

External links

E.g. links to JIRA implementation and QA tasks, and other relevant external artifacts.