



[Course](#) > [Week 3...](#) > [3.2 Thr...](#) > 3.2.2 T...

Audit Access Expires Jun 27, 2019

You lose all access to this course, including your progress, on Jun 27, 2019.

Upgrade by Jun 20, 2019 to get unlimited access to the course as long as it exists on the site. [**Upgrade now**](#)

3.2.2 Three-Part User Story Summary Points

There are three parts to a User Story:

- Value Statement
- Assumptions
- Acceptance Criteria

These work together to form the complete "User Story." It is NOT just the Value Statement.

The proper means of creating a Value Statement is up for debate, but usually the statements are as follows:

As a...[Who]...I want to...[What Functionality Desired]...in order to...[Why It's Important].

The "Who" is the user, either directly using or consuming outputs from the product the Development Team is building. It's important that if this user is not clearly defined already in the Project Charter then they are added by the Product Owner, with consensus by the Stakeholders. This ensures there's a clear understanding when say the term "End User" is put into the [Who] part of the Value Statement. Usually "End User" is too vague. It should be a descriptive title such as "Research Librarian" or "Fighter Co-Pilot."

The "What" should be the MOST important aspect of the component being built. Often because product features serve multiple users and needs or wants of users the User Stories can seem to compete with one another. When multiple types of value or outputs are created by a new product feature or component, make sure that the one that is put in the Value Statement is the priority. Priorities matter because a product that does everything for everyone in the end serves no one well.

Most importantly the "What" or "Functionality Desired" should be stated in only business terms. Examples:

- ***I want to Login Using My User Name and Password --- WRONG!***
- ***I want to access my account --- RIGHT!***

The "Why It's Important" is a critical failing of most User Stories. This is true for experienced and beginning Scrum teams. The easiest trap is to simply restate the "What" in another way. Such as "I want to Login to access my account." As stated above, the proper "What" in this statement is "access my account." So what's the "Why?" It could be:

- To check notifications
- To assign work
- To check the rankings of my fantasy football team

The key is that it's important and the MOST important "Why" of the feature or function. Others that are less important belong in the Assumptions.

Additional key points:

- Assumptions are a bucket for everything else
 - Captures less important value created by the User Story
 - Captures detailing of Why the user story is important
 - Identifies constraints from preceding or proceeding tasks, work, components, etc.
 - Identifies all the standards, influences, reference architectures, etc.
 - Captures other reasons "Why" this story might be important
 - Can limit the Acceptance Criteria and the Value Statement - not just the Value Statement

- The only information that is not needed to be listed here, are standard procedures captured in the ***Definition of Done***.
- Acceptance Criteria are NOT restatements of the Value Statement
 - Should clearly define the primary use cases for testing
 - Must specify any performance or loading that the product increment should meet
 - Must be comprehensive in detailing all tests that will be run to close the story

Finally it's important to note that all User Stories should be modular because they capture business functionality. If written correctly these shouldn't ever be in conflict because of technical dependencies.

User Stories are owned by the Product Owner when in the Product Backlog, and then once they are moved into the Sprint Backlog they are owned by the Development Team. The Development Team can update these stories with comments, but cannot change the Value Statement, Assumption, or Acceptance Criteria without the Product Owner and Development Team Members' consent.

All User Stories are governed by the Project's ***Definition of Done***. In this way the Definition of Done is the hidden "Fourth Part" of any user story, although it is focused on process for closing a story. The Definition of Done captures all the standard requirements for completing work defined in a User Story, such as:

- Standard approvals,
- Reviews by stakeholders
- Prototyping (if required)
- Documentation (for sustainability, reporting, etc.)
- Design constraints (for compliance, integration, etc.)

In this way the Definition of Done helps to modularize the statements within an User Story; but it also provides a clear set of expectations around quality control and sustainability. These benefits are further elaborated in part in this course, and in greater depth throughout the Agile Project Management Certificate program.

[Learn About Verified Certificates](#)

© All Rights Reserved