User Story & Acceptance Criteria

# **Example of User Story**

A User Story is a description of an objective a person should be able to achieve, or a feature that a person should be able to utilize, when using a software application.

User Stories have been classically written in the following form:

***As an <actor phrase > I want <action phrase> so that <outcome phrase>***

For example:

As an Administrator, I want to be able to create User Accounts so that I can grant users access to the system.

# **Acceptance Criteria**

[**Microsoft Press**](http://www.microsoft.com/learning/en/us/microsoft-press-books.aspx#fbid=Q6iHmyUaIkW) defines Acceptance Criteria as “Conditions that a software product must satisfy to be accepted by a user, customer or other stakeholder.” Google defines them as “Pre-established standards or requirements a product or project must meet.”

Acceptance criteria define the boundaries of a user story, and are used to confirm when a story is completed and working as intended. Acceptance Criteria are a set of statements, each with a clear pass/fail result, that specify both functional (e.g., minimal marketable functionality) and non-functional (e.g., minimal quality) requirements applicable at the current stage of project integration. These requirements represent “conditions of satisfaction.” There is no partial acceptance: either a criterion is met or it is not.

Acceptance Criteria must be expressed clearly, in simple language the customer would use, just like the User Story, without ambiguity as to what the expected outcome is: what is acceptable and what is not acceptable. They must be testable: easily translated into one or more manual/automated test cases.

**Acceptance criteria\*:**

1. Define the **boundaries** for a user story/feature
2. Help the product owner answer what they need in order for this feature to **provide value** (typically these are the minimum functional requirements)
3. Help the team gain a **shared understanding** of the story/feature
4. Help developers and testers to derive **tests**
5. Help developers know when to **stop** adding more functionality to a story

**Good acceptance criteria:**

1. State an intent not a solution (e.g. “The user can choose an account” rather than “The user can select the account from a drop-down”)
2. Are independent of implementation (ideally the phrasing would be the same regardless whether this feature/story would be implemented on e.g. web, mobile or a voice activated system)
3. Are relatively high level (not every detail needs to be in writing)

# **Benefits of Acceptance Criteria**

Including acceptance criteria as part of your user stories has several benefits:

* they get the team to think through how a feature or piece of functionality will work from the user’s perspective
* they remove ambiguity from requirements
* they form the tests that will confirm that a feature or piece of functionality is working and complete.

# **Examples of User Story & Acceptance Criteria**

**Example User Story 1:**

As an Administrator, I want to be able to create User Accounts so that I can grant users access to the system.

**Acceptance Criteria** for the above User Story might look like the following:

1. If I am an Administrator, I can create User Accounts.
2. I can create a User Account by entering the following information about the User: a. Name, b. Email address, c. Phone Number d. License Number (Power/Basic/None), e. Account Status (Active/Inactive), f. Reports to (from a list of "Active" Users)
3. I cannot assign a new User to report to an “Inactive” User
4. I cannot assign a new User to report to a User if it creates a cyclical relationship (e.g., User 1 reports to User 2 who reports to User 1
5. The system notifies me that it sent an email to the new User's email address, containing a system-generated initial password and instructions for the person to log in and change their password.
6. I am able to verify with the intended recipient of the email that it was received.

**Example User Story 2:  
*As an*** internet banking customer  
***I want*** to see a rolling balance for my everyday accounts  
***so that*** I know the balance of my account after each transaction is applied

**Example acceptance criteria:**

1. The rolling balance is displayed
2. The rolling balance is calculated for each transaction
3. The balance is displayed for every transaction for the full period of time transactions are available
4. The balance is not displayed if a filter has been applied

**Example User Story 3:**

As a user of the library catalogue, I want advanced search options on the front page so that I can quickly and easily refine my search.

The acceptance criteria are written in simple language that the customer would use, just like the user story. For the example above, the acceptance criteria could include:

1. I can limit the search by format/type.
2. I can delineate the search by date range.
3. I can limit the search to publisher information such as title, author, subject, place, publisher and call number.
4. I can restrict the search to a particular website/catalogue, collection.
5. I can find advanced search options – advanced search options are carried through as filters to search results page.
6. I can filter by availability.

**Example User Story 4:**

*As a customer, I want to be able to order and pay for the book via a secure-web based form.*

For the above example, the acceptance criteria could include:

1. All mandatory fields must be completed before a customer can submit a form.
2. Information from the form is stored in the customer orders database.
3. Payment can be made via credit card (Amex, Master Card, and Visa).
4. The system shall accurately calculate and apply sales tax.
5. The system shall accurately calculate and apply shipping charges.
6. The customer shall be able to verify the accuracy of the order.
7. An acknowledgment email is sent to the customer submitting the form.
8. Protection against spam is working.

## **When are they written?**

The Product Owner (the person on your team who represents the customer) may write the acceptance criteria at the same time as they write the user stories, especially if the acceptance criteria will help the team with estimation.

They may also write the acceptance criteria when the story is picked.

Acceptance criteria will often emerge out of multiple conversations between the Product Owner and the development team when they start discussing the user story. The issues and ideas raised during these conversations can be captured in the acceptance criteria.

The only rule is that acceptance criteria have to be written before the developers start work on the user story. Remember also that the acceptance criteria may be added to or refined as the development progresses and the team learns more about what they are building.

When the developers have finished the user story, they demonstrate the feature to the Product Owner, showing how each criterion is satisfied.

The benefits of acceptance criteria include:

* focusing the team on how a feature will work from the customer’s perspective
* removing ambiguity from requirements
* forming the tests that will confirm that a feature is working and complete
* limiting the developers to adding only the functionality that the user story requires.









