Guidance Document: User Story

This guidance document explains the usage of and components of a user story.

1. Definition

A user story is a concise statement of functionality or quality needed to deliver value to a specific stakeholder. A user story describes *who* needs to do *what* and *why*. A user story serves as a placeholder for further discussions to refine the project team's understanding of the stakeholder's needs.

2. Format

The recommended format for a user story is

> As a … I want to … so that …

This format ensures that the user story focuses on the value of the solution to the stakeholder. The user story is completed as follows.

> As a: Actor
> I want to: DO SOMETHING to ACHIEVE a GOAL (usually as part of a business process)
> So that I can: <describe the user goal, user benefit, or value produced by the user>

Some examples of user stories:

- As a health care consumer, I want to view the formulary associated with the health care consumer's health plan on an electronic device so I can see the drugs that are covered under the health care consumer's benefit plan.
- As a custodian of public records, I want to retain a remittance advice so that the Department of Vermont Health Access is compliant with state record retention laws.
- As a Supervisor, I want to assign a case to an Auditor based on the Programs they work with, so that the cases can be audited within the mandated timelines.

3. Acceptance Criteria

The acceptance criteria (also called conditions of satisfaction) for a user story consists of one or more testable statements that describe how the implemented solution of a given user story can be verified as meeting the need.

Rule-oriented acceptance criteria is an industry acceptable format, though scenario-oriented (given…when…then) is preferred because it's more easily converted into test cases by businesspeople and more easily understandable outside the development team.

The recommended format for acceptance criteria is scenario oriented and written in the language of the user.

Given … when … then …

The acceptance criterion is completed as follows.

1. Given I have: A CONDITION IS MET or SYSTEM INPUT or DESIGN SPECIFICATIONS
2. When I: <describe how the user is supposed to interact with the system>
3. Then I will be able to have, done, or received: EXPECTED RESULT or BENEFICIAL OUTCOME

Some examples of acceptance criteria:

- Given that I am a custodian of public records,
  when a remittance advice is received that was not migrated from OnDemand to OnBase, then the solution retains the claim for ten years.

- Given that I am a supervisor,
  when I assign a single case to an Auditor, then the case is correctly assigned to the Auditor based on the program.

- Given that I am a health care consumer, and I am logged in to an approved third-party application on a supported device,
  when I select to modify/revoke consent to see the consumer's specified health information with another individual, then that revoked/modified consent status is available to inform subsequent access by that other individual.

4. Epics, Features, and User Stories

First, it should be recognized that there is no universally accepted methodology for using epics and features within an agile framework. Epics and features provide a way of organizing user stories within a project. The EPMO approach places the epic as the highest-level artifact. An epic is broken into one or more features. Features are broken into one or more user stories.

Microsoft's Azure DevOps tool describes epics and features as follows.

> An epic represents a business initiative to be accomplished. A feature typically represents a shippable component of software. An epic represents a business initiative to be accomplished. Here are a few examples of each.
>
> Epics

- Increase customer engagement
- Improve and simplify the user experience
- Implement new architecture to improve performance
- Engineer the application to support future growth
- Support integration with external services
- Support mobile apps

Features

- Add view options to the new work hub
- Add mobile shopping cart
- Support text alerts
- Refresh the web portal with new look and feel

An epic or feature may also be written as a high-level user story. These high-level user stories are broken down into discrete user stories during project execution, typically during backlog refinement sessions.

- As a CVO Specialist or CVO Manager,
  I must be able to view Reports & Dashboards based on folder permissions,
  So that I can view insights from the data available in the system.

5. Attributes

The EPMO mandated tool to manage requirements is Microsoft Azure DevOps (ADO), the only exception to this rule is current pocket of use of Jira or ALM. (For more information on how to use Azure DevOps, including best practices, definitions, and templates, refer to the EPMO Use of ADO User Guide.) A user story should contain the attributes listed in this section.

5.1. User Story ID

The user story ID is a unique identifier.

5.2. Source/Subject Matter Expert

The name or role of the person who provided the user story.

5.3. Actor

The role or stakeholder from whose perspective the story is told.

5.4. Narrative

The statement of functionality or quality needed to deliver value to the Actor. The narrative takes the form “As a … I want to … So that …”. As an example,

> *As a data manager, I want to monitor details of data warehouse user activity, so that I can ensure the activities conform to applicable agency, state, and federal regulations.*

5.5. Acceptance Criteria

The statement of the conditions of satisfaction, e.g., how will the actor validate that the need has been met. The narrative takes the form “Given … When … Then …”. As an example,

Given that I am a data manager, when I request an account of user activities within the solution, then the solution provides details of the users and their activities based on the request.

A user story may have multiple acceptance criteria statements.

5.6. Assumptions

Any conditions relevant to the user story that are believed to be true throughout the implementation of the solution.

5.7. Comments

Any supporting information.

5.8. Value

This field can be used to align the user story with business/stakeholder requirements or technical/nonfunctional requirements. Use a value such as “business” for the former and “architectural” for the latter.

5.9. State

This field is used to track the status of the user story as it moves from a draft state until it is ready for implementation. The available states may vary depending on the project or methodology. A typical state lifecycle is, e.g., draft > verified > validated > approved.

5.10. Revised By

This field is used to track the person who has made the most recent revision to the user story.

5.11. Revision Date

This field is used to track the date of the most recent revision to the user story.

5.12. Iteration/Sprint

This field is used to record the expected iteration, sprint, release, or other implementation cycle in which the user story will be met.

5.13. Epic/Feature

These fields are used to track the upstream traceability of the user story to related features and/or epics. Record the ID of the epic/feature.

5.14. Related User Stories

This field is used to track the traceability to any related user stories. Record the ID of the related user stories.