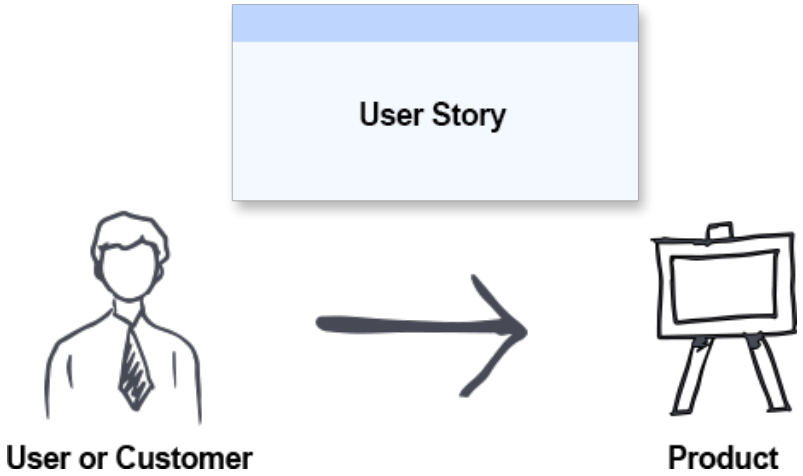
User Stories

41492 – Engenharia de Software, Nuno Sá Couto e Rafael Direito
October, 2nd 2023

User Story



- A user story is a **tool used in Agile software development** to capture a **description of a software feature from an end-user perspective**;
- A user story describes the **type of user**, **what they want** and **why**;
- A user story **helps to create a simplified description of a requirement**.

What is a User Story

- **Smallest unit of work in an agile framework**;
- **Describes an end goal expressed from the software user's perspective;**
- **Informal and general** explanation of a software feature;

Title:	Priority:	Estimate:
User Story: As a [description of user], I want [functionality] so that [benefit].		
Acceptance Criteria: Given [how things begin] When [action taken] Then [outcome of taking action]		

- The purpose of a user story is to articulate how a piece of work will deliver a particular value back to the customer;
- User stories are a few sentences in simple language that outline the desired outcome:
  - **They don't go into detail;**
  - Requirements are added later once agreed upon by the team.

Why writing a User Story

- **Stories keep the focus on the user**
  - A to-do list keeps the team focused on tasks that need to be checked off, but a collection of stories keeps the team focused on solving problems for real users.

- **Stories enable collaboration**
  - With the end goal defined, the team can work together to decide how best to serve the user and meet that goal.

- **Stories drive creative solutions**
  - Stories encourage the team to think critically and creatively about how to best solve for an end goal.

- **Stories create momentum**
  - With each passing story, the development team enjoys a small challenge and a small win, driving momentum.

How to write a User Story

Consider the following when writing user stories:

- **Definition of “done”**
  - The story is generally “done” when the user can complete the outlined task, but make sure to define what that is.

- **Outline subtasks or tasks**
  - Decide which specific steps need to be completed and who is responsible for each of them.

- **User personas**
  - For whom? If there are multiple end users, consider making multiple stories.

How to write a User Story

Consider the following when writing user stories:

- **Ordered Steps**
  - Write a story for each step in a larger process.

- **Listen to feedback**
  - Talk to your users and capture the problem or need in their words. No need to guess at stories when you can source them from your customers.

- **Time**
  - Time is a touchy subject. Many development teams avoid discussions of time altogether, relying instead on their estimation frameworks. Since stories should be completable in one sprint, stories that might take weeks or months to complete should be broken up into smaller stories or should be considered their own epic.

INVEST

A good User Story should be INVEST

- **Independent:** Should be self-contained in a way that allows them to be released without depending on one another.
- **Negotiable:** Only capture the essence of the user's need, leaving room for conversation. User stories should not be written like contracts.
- **Valuable:** Delivers value to end user.
- **Estimable:** User stories have to be estimated so they can be properly prioritized and fit into sprints.
- **Small:** A user story is a small chunk of work that allows it to be completed in about 3 to 4 days.
- **Testable**: A user story has to be confirmed via pre-written acceptance criteria.

User Story Template

User stories are often expressed in a simple sentence, structured as follows:

As an Internet banking customer
I want to see a rolling balance for my everyday accounts
So that I can keep track of my spending after each transaction is applied

User Story Template

As an Internet banking customer
I want to see a rolling balance for my everyday accounts
So that I can keep track of my spending after each transaction is applied

As a social media enthusiast,
I want to be able to schedule and automate my posts
So that I can maintain an active online presence without constantly being online

As a solo traveler
I want a feature that connects me with like-minded travelers or local guides
So I can meet new people and have a more social and immersive travel experience

As a budget-conscious traveler
I want an app section to compare prices for flights, hotels, and transportation options
So that I can find the best deals and save money on my travels.

All these User Stories map Functional Requirements

How to map Non Functional Requirements ???

Functional vs Non-Functional Requirements

- **Functional Requirements:** Functional requirements are the specifications of the product's functions (features). Simply put, functional requirements define what precisely a software must do and how the system must respond to inputs.
    - **Example:** In an e-commerce website, a functional requirement might be "The system must allow users to add products to their shopping cart

- **Non-Functional Requirements:** Non-functional requirements are a set of specifications that describe the system's operation capabilities and constraints and attempt to improve its functionality.
    - **Example:** The e-commerce website should have an availability of 99.99%

Non-Functional Requirements – User Stories

- A common challenge for an Agile team is dealing with capturing non-functional requirements in a user story.
- **The non-functional requirements can be written as a user story** and made available in the product backlog or sprint backlog.

    - Example: "**As an e-commerce website user, I want the website to be available 99.99999% of the time, so that I purchase products whenever I feel like it.**"

- **NFR can also be included as Acceptance Criteria in a user story.**

- For example, for the following user story, "As an e-commerce website user, I want to search for products so that I can purchase them". **NFR as acceptance criteria would be "application responds to search request within 5 seconds from the time request is received".**

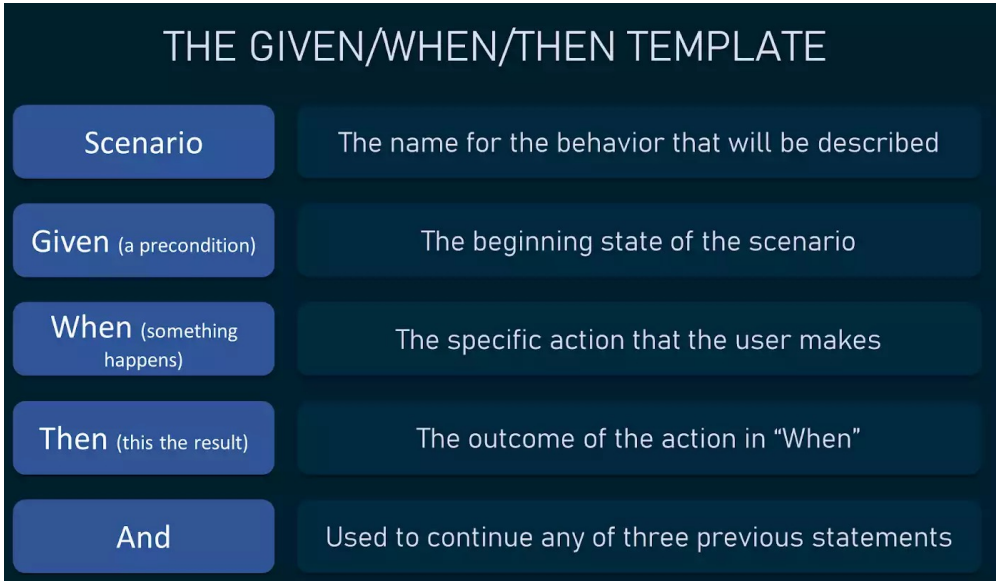
Acceptance Criteria

- Acceptance criteria are the **conditions that a software product must meet to be accepted by a user, a customer, or other systems**.
- **They are unique for each user story** and define the feature behavior from the end-user's perspective.
- **Well-written acceptance criteria help avoid unexpected results at the end of a development stage and ensure that all stakeholders and users are satisfied with what they get.**
- There is no allocated responsibility for writing acceptance criteria.
    - While it is usually the product owner or product manager who gets around to defining the functionality, just about anyone on the team could write acceptance criteria for user stories.

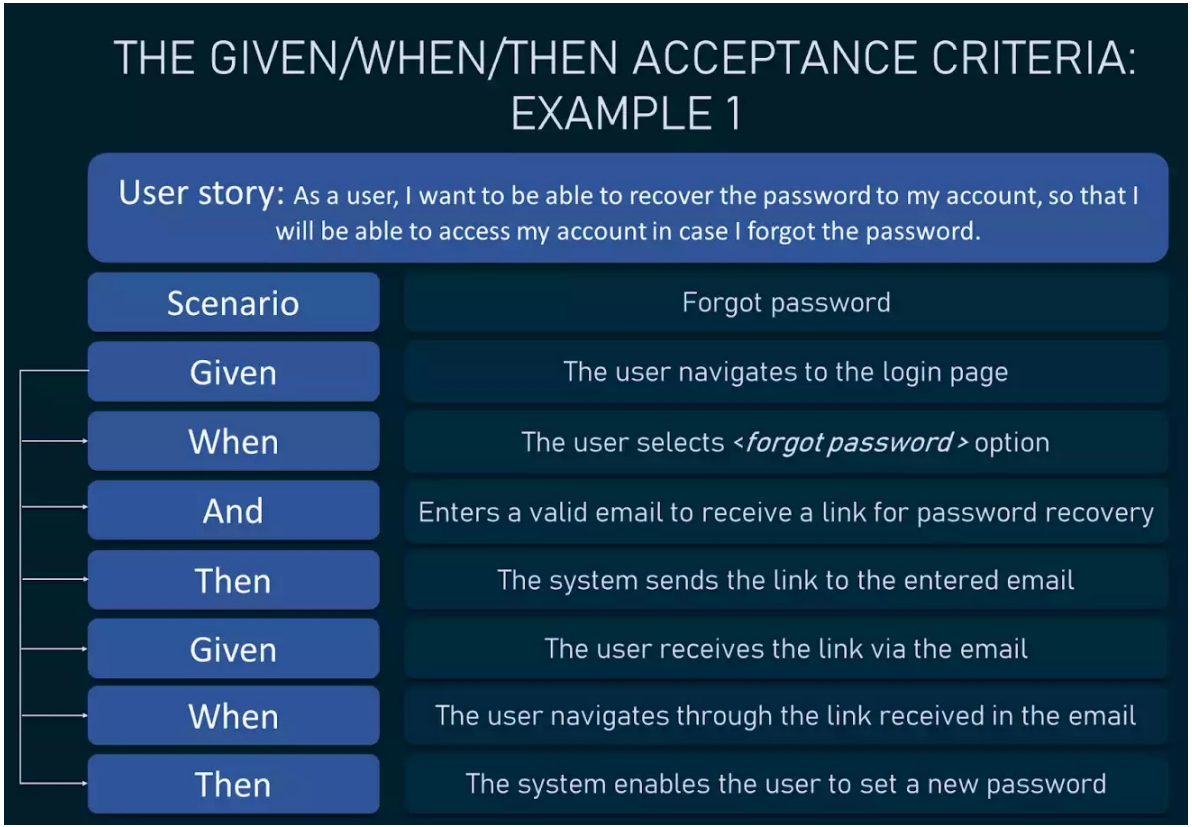
Scenario-Oriented Acceptance Criteria

- The scenario-oriented **format is the acceptance criteria type that comes in the scenario form and illustrates each criterion**.

- It is approached through the **Given/When/Then (GWT) sequence** that looks like this:



Scenario-Oriented Acceptance Criteria



Definition of Ready vs Definition of Done

The key difference between the definition of ready and the definition of done is that:

- **the definition of ready covers the requirements coming into the sprint.**
- **the definition of done covers the product coming out of the sprint.**

- **The definition of ready (DoR) applies to your user stories.**
    - It makes transparent your team's shared understanding of what's needed for a user story to be brought into a sprint.

- **The definition of done (DoD) applies to your working software.**
    - It makes transparent your team's shared understanding of the quality standards a piece of work needs to reach to be releasable.

Definition of Ready - Example

Stories are:

- Independent
- Negotiable
- Valuable
- Estimable
- Small
- Testable
- Prioritized
- Set up to provide all content, designs and assets
- Understood by the whole team, including risks, assumptions or constraints
- Shared with any external contributors or reviewers to book in their availability

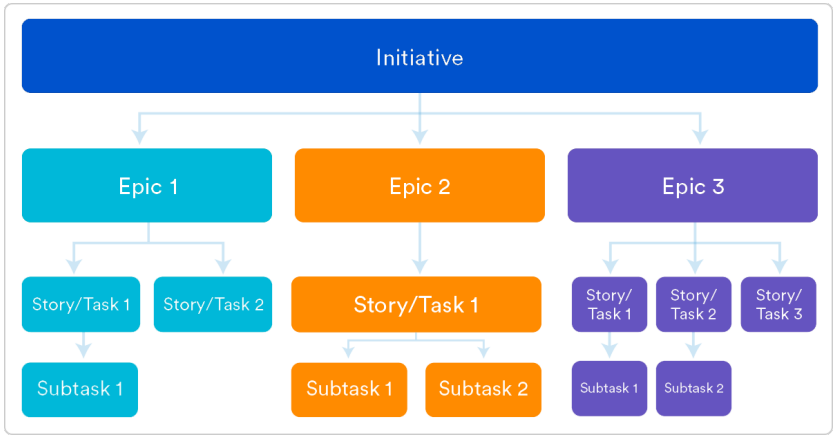
Definition of Done - Example

For a software project, here's what your DoD might include:

- Tests written and passing
- Security vulnerability scan passing
- Continuous Integration build passing
- Cross browser testing done on current top 5 browsers according to analytics
- Mobile testing done on current top 3 mobile devices according to analytics
- Google accessibility check passed
- Code peer-reviewed
- Documentation updated
- Acceptance criteria met

Initiatives, Epics, and User Stories

- A **user story** (or simply – story) is a **brief request or requirement composed from the end users' perspective.**
- An **epic** is a **large piece of work that is typically broken down into small pieces** (tasks known as stories).
- An **initiative** represents **epic collections that lead to a common goal**.



Initiatives, Epics, and User Stories

INDUSTRY	INITIATIVE	EPIC	USER STORY
Banking / Software development	Mobile banking	• Checking account • Savings account • Company account	As a customer, I want to check my account balance, so that overdrafts don't happen.
Marketing	Product launch campaign	• Internet campaign • Tv campaign • Print media advertising	• Design Internet ads, • Create a TV advert • Media planning
Retail	Introduce regional products to a supermarket chain	• Cheeses range • Smoked meats range • Deli range	• Find suppliers, • Verify quality, • Test-launch new assortment at selected locations
Tourism, travel agency	Set foot in Australia	• Select hotels • Hire guides • Obtain licenses	• Research hotels in Victoria • Research hotels in Tasmania • Interview candidates by state
Approximate duration	Several months to a year to complete.	Epics are to be just manageable (the right number of stories in an epic).	One sprint.

BigPicture

Who Writes User Stories

Who writes user stories? **Anyone can write user stories.**

Does the Product Owner write User Stories?

- It's the product owner's responsibility to make sure a product backlog of agile user stories exists, but **that doesn’t mean that the product owner is the one who writes them**.

- Over the course of a good agile project**, you should expect to have user stories written by each team member**.

Also, note that **who writes a user story is far less important than who is involved in the discussions of it.**