**USER STORIES**

WHAT ARE USER STORIES?

It is the smallest unit of work to be done which carries a certain business value with it.

A user story created to capture a product functionality from user’s perspective.

WHY CREATE USER STORIES?

* **Stories keep the focus on the user:** 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.

WORKING WITH USER STORIES

1. Once a story has been written, it’s time to integrate it into the workflow.
2. During a sprint or iteration planning meeting, the team decides what stories they’ll tackle that sprint. Teams discuss the requirements and functionality that each user story requires.
3. Another step in this meeting is to score the stories based on their complexity or time to completion.
4. Teams use t-shirt sizes, the Fibonacci sequence, or planning poker to make proper estimations.

The most commonly used standard format for a User Story creation is stated below:

**As a <user role/customer, I want to < goal to be accomplished> so that I can <reason of the goal>**

Example:

As a WhatsApp user, I want a camera icon in the chat write box to capture and send pictures so that I can click and share my pictures simultaneously with all my friends.

3 C’S OF USER STORY

* CARD: Cards are where User stories are written. A card forms the basis for a conversation. When a card is placed om the board, the team is committed to discussing that specific user requirement. The cards are usually 3 inches by 5 inches.
* CONVERSATION: Once the user stories are written, they are then opened to have more discussions, conversations around these requirements. Many meetings like backlog grooming, sprint planning, release planning etc. form the stage for these conversations and information exchange on user stories. The focus is to have more of ongoing conversations and less of detailing the requirements upfront.
* CONFIRMATION: The C is to build user stories acceptance criteria.
  + Acceptance Criteria: Acceptance Criteria of any user story are the conditions that should be fulfilled to satisfy the user or product owner before marking the user story complete. Teams have the acceptance criteria to be fulfilled as their definition of done.

These are the conditions of contentment on the user story card. They can be written on the back side of the card.

The format for acceptance criteria is as follows:

“Given some precondition when I do some action then I expect the result”.

**Example (w.r.t for the above user story):**

* Please consider chatting with a friend and I should be able to capture a picture.
* When I click on a picture, I should be able to add a caption to the image before sending it.
* If there is some problem with starting my phone camera, an error message like “Camera could not be started.” etc., should be shown accordingly.

Hence, the User story defines the requirement for any functionality or feature while the Acceptance Criteria defines the “Definition of done” for the user story or the requirement.

WHO CAN CREATE A USER STORY?

1. Developer.

2. Team member.

3. product owner.->ownership will be with him

4. Some times ->scrum master.

Acceptence criteria come up with the team, product owner decide to priority or depriortise. So, acceptence criteria is with product owner.

**Case 1:**

Before 3 years, I was working on a Mobile Application Project and the product was an application that was designed for the delivery people.

You would have seen a delivery person coming to your place for delivery. They have a mobile phone on which they ask you to give your signature after delivery. This signature reflects on the portal of the courier service providers like DTDC, FedEx etc.

*Let’s imagine that the mobile app is just launched and their portals are already existing and up.*

**Problem:** For a Sprint your Product owner has a user story for this mobile app that “As a Portal Admin, I should be able to view the signature taken by the delivery person at the time of delivery”. Here the portal (web app) has been changed and updated accordingly to reflect the signature.

As a QA you have to verify if the signature captured in the mobile app is reflecting as expected in the portal.

If you look at this user story, it looks simple, but there is a hidden requirement here that “For historical deliveries, there was no signature reflection functionality, so what should happen if the portal guys view historical deliveries?” Should historical data be wiped out? Should we allow crashes or errors for such data?

Of course not at all, this should be handled graciously.

**Solution:** When the respective DB tables are updated to add a new column for the Signature location, the old data should have a NULL or 0 value which should be checked and a message stating “No signature exists” should be shown.

This can be called a miss by the Product Owner or Business Analyst, but this has to be done. Implementing one feature successfully but breaking something along with it is not desirable by the customers. This needs to be done along with the same user story and in the same sprint.

WHAT IS AN AGILE EPIC?

An epic is a large body of work that can be broken down into several smaller stories, or sometimes called “Issues” in Jira. Epics often encompass multiple teams, on multiple projects, and can even be tracked on multiple boards. Epics are almost always delivered over a set of sprints.

EPICS, AND INITIATIVES?

* User stories are also the building blocks of larger agile frameworks like epics and initiatives.
* Epics are large work items broken down into a set of stories, and multiple epics comprise an initiative.
* These larger structures ensure that the day-to-day work of the development team (on stores) contributes to the organizational goals built into epics and initiatives.

A diagram of a project

Description automatically generated with medium confidence

Example:

As a restaurant customer, I want to reserve a table online, so I can ensure I have a place to eat at my preferred time.

Acceptance criteria: The system should show available times for reservations. After reservation, the customer should receive a confirmation.

Four main types of user stories:

* Simple: These are individual or self-contained user stories that focus on a particular user or type of product.
* Epic: Groups of related user stories come together to form epics. They may involve multiple users working together or independently or multiple needs for a single type of user to achieve some goal or benefit.
* Thematic: These are major investments and strategies that group epics together. Thematic user stories highlight how a company will achieve wider goals.
* Scaled Agile Framework (SAFe): These user stories add extra details such as a benefit hypothesis, cost of delay, or nonfunctional requirements.