User Stories

08 March 2019 14:02

What is a User story?

- Product backlog is set of simple requirements.
  - Not a big document
  - Not detail oriented.

  - Product back log is built using set of User stories
  - User story is a feature/requirement from a user perspective – Simple story cart explaining requirement in simple words.
  - Example:
    - **As a** user
    - **I want** a news age
    - **So that** I can read top news stories
  - This
    - Reduces documentation and
    - Allows customer collaboration
    - Upfront details aren't necessary
      - We don't have to care about a particular feature until we start building that feature or part of the project.
        - But what if we get certain details here that might change requirement on the other user story or requirement?
          - Answer: It could happen – Only way to reduce it is to be careful.

How to build User stories?

- By following a principle called INVEST – Very important before writing a User story
  - I – Independent – Making sure each User story is self-contained or independent from other User stories.
  - N – Negotiable – It basically means even the product owner is open in a discussion where User stories are discussed and decided – this continues to apply during User story prioritization. Everybody listens to everyone.
  - V – Value – Making sure each User story is valuable to the business/end customer. Otherwise, there is no point putting time and effort in that User story. Nothing is done unless it gives value.
  - E – Estimable – Every User story is written in a way that team can read it, understand it and put an estimation on it. There should be enough information to estimate the time it takes to complete.
  - S – Small – Dividing the project in the form of small chunks that adds value to the customer/business.
  - T – Testable – Making sure each piece of work is testable and is tested before concluding it's done.

  Another principle:
  CCC
- C – Card – Information is captured in this cart along with acceptance criteria.
- C – Conversation – Card is used as a topic of discussion or conversation between team-team member or team – product owner
- C – Confirmation – Once the task is done in team's perspective, Product owner confirms whether the talk is done.

User story:

- User stories are written on piece of papers(cards)
- Back of each card contains Acceptance criteria – Product owner defines/articulates them in the presence of a QA/tester.
  - Articulate when the user story is "done" from the product owner's perspective (On behalf of the user)

        - Forms "acceptance tests" that a quality assurance tester can use to verify quality.
        - BA(business analyst might help here heavily)
    - Plus Edge cases.

As a **<Role>**
I want **<Requirement>**
So that **<Reason/ ROI – Return on investment>**

F1 Car game example:
As a **F1 game player**
I want **to see a circuit map**
So that **I know where I am on the circuit**

Acceptance criteria template:
A <Role> should be see/be able to:
- **AC(Acceptance Criteria) 1**
- **AC 2**
- **Etc**

Edge cases(other scenarios):
- **EC(Edge Case) 1**
- **EC 2**
- **Etc**

These logical criteria that can be used to test/verify whether a "User story" is complete.

Example:
Acceptance criteria example:
An F1 game player should be able to see a track that:
- Is as closely as resembles the shape of the shape of the F1 track in the game.
- Consists of a geographical circuit line that fits within a 3 by 5 proportional/ratio box.
- Has a small dot to represent the game player's car
- The dot should move in relation to and in proportion to the game players position on the track

Edge case(s)
- If the car blows up, the dot should turn red for 1 second and then disappear

- **Based on these inputs, designers work on the exact position, look and feel and developers implement them.**
- **Product owner works with the team to fine-tune, clarify and confirm stories.**

Epics:

What is an Epic?
Epic is nothing more than a large Story

Example:
As a **F1 game player**,
I want **an option screen**
So that **I can choose the driver(avatar/character) I want to use**

The requirement here is an **Option screen** which may consists of many other settings/options for the user. At this point, it is important to dig deeper into what other options are need to be added on to the **options screen**. Hence, building an **options screen** becomes an Epic and not an actual User story for the team(design and developers). This epic should be further divided into smaller user stories.

So, what is an Epic?
- A large story
- Something that cannot be completed within an iteration(sprint)
- Needs to be bracken down into separate User stories before working on it.

A Theme

A theme is a collection of stories with shared attributes.

An example:
As a **Sports fan**
I want **a group of sports pages**
So that **I can see the top 10 news stories for each sport**

These are generally used to refer to group of stories which share a common function/attribute

So, based on above example, it is obvious that, we can divide the theme into more stories, like so:

Sports home page
As a Soccer fan
I want a Football home page
So that I can see the top 10 football news stories

Sports home page
As a Baseball fan
I want a Baseball home page
So that I can see the top 10 Baseball news stories

** In this example, only 2 sports are considered. Theme could be more than just 2 stories*

Difference between Epic and theme are:

- Even though both consists of multiple stories within them,
  - Stories within Theme has common attribute
  - Whereas in Epics, stories are mostly independent

- In case of Theme, it is obvious that it has multiple stories
- In case of an Epic, it is not obvious at the initial stage how many stories could eventually come under it and it is hard to estimate how large each story can be within an Epic.

Note: A good way to tackle a T**heme** is to work on *just one* **User story** of that theme; once that is done and once we have all the details needed to work on single story, it will be easy to work on other stories of that theme because other stories got the same *theme(attributes).*

When to Split stories:

- When it cannot be completed in an iteration/sprint
- When it's hard to get an estimate

How to split stories?

- Avoid meeting non-functional requirements
  - Don't meet performance constraints
    - Example: The user should be able to load the page in less than 30ms(just an example).

Example:
Acceptance criteria:

- The user should be able to load the page in less than 30ms(just an example)
- If there is an unknown error loading the page, the user should see the standard error

        - This could be a separate story/requirement
- (Remove) Cross cutting concern
    - Usually it is a piece of acceptance criteria/requirement which apples to more than one story.
        - Example: If there is an unknown error loading the page, the user should see the standard error message - "Unknown error: Currently unable to load this page. Please refresh the page and try again."
            - The error message should be added to other pages as well – Hence this could be a separate story on its own.
- (Separate by) Mixed priority
    - It is important to think back the priority of piece of work.
        - The user should see an advertisement for the sports channel below each news story , measuring 5 by 2 inches
            - The product owner and the team could ask themselves, do we really need to complete this piece of work at this point in time?
            - If not, this could be a separate story on its own in the future sprints – even at the end of completion of functionality work if not next sprint.
        - Also, if the task is hard/tricky to estimate, or the criteria doesn't fit in an iteration, it is worth asking the question.
            - It could be that, there is no real business need to deliver by this piece of functionality/feature at the present time. It can be delayed for further iteration.

message - "Unknown error: Currently unable to load this page. Please refresh the page and try again."
The user should see an advertisement for the sports channel below each news story , measuring 5 by 2 inches