User Story Research

What are user stories?

User stories are a key component in creating an Agile driven software development team. User stories give context, in non-technical language, to the developers how end users can use the application. They are not requirements or features that need to be build, they're overarching goals.

Once a user story is written, the team can come together and set-up requirements for this user story. What is needed to finish this story? When a project has multiple stories worked out with requirements, it becomes easier for teams to estimate how much time will be needed to finish a story. These user stories fit together in a larger plan to help manage large projects.

Stories often focus on one small part of a project. With multiple small stories you can work to completing a bigger, overarching goal, called an epic. An epic is a larger body of work that can be split up in several smaller user stories. Where the idea of a user story to be a small workload that can be completed within a one- or two-week sprint, epics take longer. Generally, epics are scheduled over a larger period, such as a quarter.

While epics are large bodies of smaller tasks, even they have an overarching goal to work to: initiatives. An initiative consists of multiple epics and can take up to multiple quarters or a full year to complete.

The benefits of user stories

While it might sound easier to just dissect an epic into requirements and work on those, having user stories has an added benefit over raw requirements.

Working with user stories keeps the team focused on what matters most, the users. As the word says, users are at the forefront of these tasks. The team is always working and thinking from the perspective of the user.

As mentioned earlier, these stories are written in a simple, non-technical language. This means anyone can work, create, and understand these user stories. It allows users, other team members, and stakeholders to all understand what the team is currently working on.

User stories are often accompanied by prioritization, allowing stakeholders to determine which features are most important. This prioritization helps in planning iterations and releases, ensuring that the highest-value user stories are implemented first.

User stories promote adaptability and flexibility in the development process. As new information emerges, priorities change, or user needs evolve, user stories can be reprioritized, added, or modified to reflect the latest understanding and ensure the software aligns with the current requirements.

Writing user stories

When writing user stories, there are guidelines on what they should generally look like.

The first step to creating a story is thinking who it's for. It's simple to say "the customer" in a story, but it can be more specific. User stories are all about the user, so really knowing who you're making this story and application for can help make the stories more specific.

Define when a user story is considered done. What needs to be completed for this user story to be finished? This furthers the clarity of what the team is working on, ensuring everyone is working towards the same goal.

To reach the point of marking a story as done, the team needs to know what to do with them. This can be done by creating small, but important sub-tasks that are required to mark this story as done. You can assign members of the team to a sub-task. Together the team can then complete a story.

You can make sure you're writing good user stories when they meet the *INVEST* criteria:

- Independent
- Negotiable
- Valuable
- Estimable
- Small
- Testable

Writing user stories with these criteria in mind will ensure you are writing efficient and optimal user stories.

Examples

To give myself some practice, I am going to write some user stories that could be implemented in my personal project. I will also explain where I think I'm using INVEST or the guidelines mentioned above.

1. As an 'eSports fan' I want 'a place where I can see upcoming matches' so that 'I know when my favourite team is playing'.

Independent: The story does not rely on other functionality of the application. A single location where matches are shown would be enough for this story.

Negotiable: While we know it should probably all be in one single location, we have not specified where this location is, what it should be or look like. There's room to discuss this with the stakeholder and user feedback.

Valuable: The story addresses a need for eSports fans who want to stay informed about upcoming matches. By providing a centralized place to see this information, fans can easily keep track of when their favourite teams are playing, which enhances their overall experience.

Estimable: The story is clear and precise, making it relatively easy to estimate the effort required to implement it. The team can break it down into smaller tasks, such as designing the interface, integrating with match data sources, and implementing notification features.

Small: The user story is small enough to be implemented within a single sprint or development cycle. It has a clear scope and can be completed in a reasonable amount of time, allowing for faster feedback and iteration.

Testable: The user story has a clearly defined outcome that can be tested. The success criteria for this user story can be measured by verifying that the user can access the "place" and view the upcoming matches, ensuring the information is accurate and up to date.

2. *As an 'administrator' I want 'functionality to add new matches' so I can 'keep the users informed on upcoming matches.'*

Independent: The story focuses on a specific feature requested by the administrator. It does not rely on other user stories or functionalities to be implemented. The functionality to add new matches can be developed and delivered separately without being tied to other system components.

Negotiable: The user story allows for negotiation and flexibility in its implementation. The specific design and implementation details of the functionality to add new matches can be discussed and adjusted based on the administrator's requirements, system constraints, and stakeholder feedback.

Valuable: The story addresses the needs of the administrator who wants to keep the users informed about upcoming matches. By providing the functionality to add new matches, the administrator can ensure that the system's match information remains up-to-date and accurate, thereby enhancing the user experience and keeping them well-informed.

While this ties in with the previous story, neither hold one back. Both can work without the other, but they provide value for each other.

Estimable: The story is estimable as it clearly defines the desired functionality - the ability to add new matches. The development team can estimate the effort required for implementing this feature by breaking it down into smaller tasks, such as designing the user interface for adding matches.

Small: The story is small in scope, focusing on a specific functionality rather than encompassing multiple features. The development team can probably implement this user story within a reasonable timeframe.

Testable: The story has a clearly defined outcome that can be tested. The story is done when it can be validated by verifying that the administrator can successfully add new matches to the system, ensuring that the information is stored correctly and can be displayed to the users accurately.

Used:

https://www.atlassian.com/agile/project-management/user-stories

https://www.atlassian.com/agile/project-management/epics-stories-themes

https://stormotion.io/blog/how-to-write-a-good-user-story-with-examples-templates/#how-to-write-user-stories