User Story

A user story is a short, informal description of a software feature written from the perspective of the end user. It describes the user's needs, goals, and the reason why the feature is desired. User stories are used in Agile software development to capture requirements and facilitate collaboration between the development team and stakeholders. They are typically written in a simple format, such as "As a <type of user>, I want <some goal> so that <some reason>." User stories are not detailed technical specifications but rather serve as a reminder for future conversations about the feature. They help to shift the focus from writing about requirements to discussing them, encouraging collaboration and creative solutions.

A user story is a concise, informal description of a feature or functionality from an end-user perspective,

typically used in agile software development to capture requirements.

It focuses on what the user needs to accomplish and why, rather than detailed technical specifications. Each user story follows a specific format:

A user story typically follows a structured format that includes:

Role: The type of user or persona involved.

Goal: What the user wants to accomplish.

Reason: Why the user wants to accomplish this goal

.

Here's an example of Users story-

"As a [role], I want to [goal] so that [reason].

For instance:

"As a student, I want to check my grades online so that I can track my academic progress easily."

In this user story:

Role: Student

Goal: Check grades online

Reason: To track academic progress easily

This user story helps the development team understand that students want convenient access to their grades

online to stay updated on their academic performance.

Example 2-

As a frequent traveler, I want to receive real-time notifications about flight delays so that I can adjust my travel plans accordingly."

Role: Frequent traveler

Goal: Receive real-time notifications about flight delays

Action: Receive push notifications on my mobile device when there is a delay or change in my flight schedule.

Benefit: Allows the user to stay informed and make necessary adjustments to their travel plans in a timely manner.

**Acceptance Criteria:**

Notifications should be sent immediately upon detection of a flight delay or change.

Notifications should include details such as the flight number, departure/arrival time, and reason for the delay.

Users should have the option to view additional information and take appropriate actions directly from the notification (e.g., rebooking options).

The system should be able to handle a high volume of notifications without delays or errors.

# How to create a good story

"Invest" in the context of Agile methodology, particularly in Agile software development, refers to the acronym INVEST,

which outlines criteria for writing effective user stories. Each letter stands for a characteristic that a good user story should possess:

I: Independent: User stories should be self-contained and not dependent on other stories.

N: Negotiable: Details of the user story can be negotiated between the product owner and the development team.

V: Valuable: Each user story should deliver value to the user or the business.

E: Estimable: The development team should be able to estimate the effort required to implement the user story.

S: Small: User stories should be small enough to be completed within a single iteration or sprint.

T: Testable: The user story should have clear acceptance criteria that define when the story is completed and working as expected.

Adhering to these principles ensures that user stories are well-defined, manageable, and contribute to the overall success of the project.

Users Story- user story for an e-commerce platform:

"As a customer, I want to be able to track the delivery status of my order so that I can anticipate its arrival."

Role: Customer

Goal: Track delivery status of order

Action: Access the order tracking page, enter the order number or other identifying information,

and view real-time updates on the delivery status.

Benefit: Allows customers to stay informed about the progress of their orders and plan accordingly for delivery.

**Acceptance Criteria:**

The e-commerce platform should provide a dedicated section or page for order tracking accessible from the customer account dashboard.

Customers should be able to easily locate the order tracking feature and enter the necessary information,

such as order number or email address associated with the order.

The system should display relevant details about the order status,

including shipping date, expected delivery date, current location (if available),

and any delivery updates or notifications.

The order tracking information should be updated in real-time to provide accurate and timely status updates.

If the order is out for delivery, customers should be provided with estimated arrival time windows or live tracking options, if available.

This user story focuses on enhancing the customer experience by providing transparency and visibility into the delivery process,

ultimately improving customer satisfaction and loyalty.

**INVEST**

The INVEST principle in user stories stands for Independent, Negotiable, Valuable, Estimable, Small, and Testable. This principle helps ensure that user stories are well-defined and meet certain criteria to be effective in Agile software development. Here is a breakdown of what each aspect of the INVEST principle entails:

• **Independent:** User stories should be conceptually separate from other stories and not reliant on the completion of others.

• **Negotiable**: They should prompt discussion but not prescribe a solution, allowing for evolving conversations between the product owner and development team.

• **Valuable**: User stories need to explicitly state the value they bring to users, ensuring they deliver tangible benefits.

• **Estimable**: They should provide enough detail for the development team to estimate their size accurately.

• **Small**: User stories should be the smallest piece of work that delivers useful software, aligning with Agile's short iteration cycles.

• **Testable**: They must be clear enough to assess if they are done, with defined acceptance criteria for verification.

By adhering to the INVEST principle, teams can create high-quality user stories that are focused, valuable, and conducive to efficient Agile development practices.

**SCRUM**

Scrum is an agile framework used for managing and organizing complex projects, primarily in software development, but it has also found applications in other fields such as marketing, research, and even education. It emphasizes iterative and incremental progress, flexibility, and collaboration among team members. Here's a detailed overview of the key components and principles of Scrum:

Roles:

Scrum Master: The Scrum Master is responsible for ensuring that the Scrum process is followed and facilitating communication and collaboration within the team. They remove any impediments that hinder the team's progress.

Product Owner: The Product Owner represents the stakeholders and is responsible for managing the product backlog, prioritizing tasks, and ensuring that the team delivers value to the customer.

Development Team: The Development Team consists of professionals who work together to deliver the product increment. They are self-organizing and cross-functional, meaning they have all the skills necessary to complete the work.

Artifacts:

Product Backlog: The product backlog is a prioritized list of all features, enhancements, bug fixes, and other work that needs to be done on the project. It is managed by the Product Owner and evolves over time.

Sprint Backlog: The sprint backlog is a subset of the product backlog selected for a specific sprint. It contains the user stories or tasks that the team commits to completing during the sprint.

Increment: The increment is the sum of all the product backlog items completed during a sprint, plus all the increments from previous sprints. It represents a potentially shippable product.

Events:

Sprint: A sprint is a time-boxed iteration, usually lasting between one to four weeks, during which the team works to complete a set of tasks from the sprint backlog.

Sprint Planning: At the beginning of each sprint, the team holds a sprint planning meeting to select the items from the product backlog that will be worked on during the sprint and to create a sprint backlog.

Daily Scrum: The daily Scrum, or daily stand-up, is a short meeting held by the development team every day to synchronize their activities, discuss progress, and identify any obstacles.

Sprint Review: At the end of each sprint, the team holds a sprint review meeting to demonstrate the completed work to stakeholders and gather feedback.

Sprint Retrospective: Also held at the end of each sprint, the sprint retrospective is a meeting where the team reflects on the sprint process, identifies what went well and what could be improved, and agrees on actions to take in the next sprint.

Principles:

Empirical Process Control: Scrum is based on the principles of transparency, inspection, and adaptation. The team regularly inspects the product and the process and adapts as necessary to optimize value delivery.

Self-Organization: Scrum teams are self-organizing, meaning they have the autonomy to decide how to accomplish their work and are accountable for the outcomes.

Iterative and Incremental Delivery: Scrum breaks down the work into small, manageable increments, allowing for continuous feedback and early delivery of valuable features.

Customer Collaboration: Scrum encourages close collaboration between the development team and the customer or product owner to ensure that the product meets the customer's needs and expectations.

Overall, Scrum provides a framework for teams to efficiently and effectively deliver high-quality products by embracing change, focusing on customer value, and fostering a collaborative and adaptive working environment.

**3 C’s in Agile**

1. Card: The "card" represents the physical or digital artifact that embodies the user story. It's typically a written description of a feature or functionality from the perspective of an end user. The card should be concise and easy to understand, containing just enough information to communicate the essence of the user story.

2. Conversation: The "conversation" emphasizes the importance of ongoing dialogue and collaboration between stakeholders, including the Product Owner, development team, and end users. Rather than relying solely on written documentation, Agile encourages face-to-face communication to clarify requirements, gather feedback, and ensure shared understanding of the user story.

3. Confirmation: The "confirmation" refers to the acceptance criteria or conditions of satisfaction that define when a user story is complete and meets the expectations of the stakeholders. These criteria are established collaboratively during sprint planning and serve as a basis for testing and validation. By defining clear acceptance criteria upfront, the team ensures that the user story delivers tangible value to the customer.

Overall, the 3 Cs framework underscores the Agile principles of simplicity, collaboration, and customer focus, guiding teams in creating user stories that are clear, actionable, and aligned with the needs of the stakeholders.