-User stories are part of an agile approach that helps shift the focus from writing about requirements to talking about them.

-Every agile user story includes a written sentence or two and, more importantly, sparks a series of conversations about the features and functionality the user story represents.

-User stories help to capture the essence of what users need or want from a product or feature

-high priority, low priority user stories

-User stories are a way to describe the desired functionality of product backlog items.

-**High-priority** user stories tend to be more detailed; **low-priority** tend to be less detailed.

-A user story is a short, simple description of a feature told from the perspective of the person who desires the new capability, usually a user or customer of the system

-Historically user stories were deliberately kept informal, written on index cards or sticky notes

-In fact, these discussions are more important than whatever text is written.

**What Is a Good User Story? (3C’s)**

Agile user stories are composed of three aspects that Ron Jeffries named in 2001

Card: Written description of the story, used for planning and as a reminder

Conversation: Conversations about the story that serve to flesh out the details of the story

Confirmation: Tests that convey and document details that can be used to determine when a story is complete.

How to write a user story?

Writing good user stories in Scrum requires an understanding of the basic user story template, a focus on the user or customer, and a clear picture of the desired functionality.

### **User Story Template**

When writing a user story, remember that user stories follow a standard template:  
**As a < type of user >, I want < some goal > so that < some reason >**

**[User Role]: [Action] [Benefit or Goal]**

**User Story1:**

As a customer, I want to be able to track my order status online so that I can stay informed about the progress of my delivery.

**Acceptance Criteria:**

1. When I log in to my account, I should see a section labeled "Order Status" on the dashboard.

2. The "Order Status" section should display a list of all my recent orders.

**User Story2:**

As a mobile app user, I want to be able to log in using my fingerprint so that I can access my account quickly and securely.

**Acceptance Criteria**:

1. The mobile app should support biometric authentication methods such as fingerprint scanning or facial recognition.
2. Users should have the option to enable biometric login during the account setup process.
3. Upon launching the app, users should be prompted to authenticate using their fingerprint or face.

**User Story3:**

As a website visitor, I want to be able to filter search results by price range so that I can find products within my budget.

**Acceptance Criteria**:

1. The search results page should include a filter option labeled "Price Range."
2. Users should be able to specify a minimum and maximum price range using input fields or sliders.
3. Upon applying the price range filter, the search results should update dynamically to display only products within the selected price range.
4. The filter should be accessible and functional on both desktop and mobile devices.

**User Story4:**

As a registered user, I want to be able to update my profile information so that my account reflects accurate and up-to-date details.

**Acceptance Criteria**:

1. The profile editing interface should allow users to update information such as name, email address, contact number, and profile picture.
2. Changes made to the profile should be saved automatically upon submission.
3. Users should receive a confirmation message or notification after successfully updating their profile.

**INVEST:**

"Invest" is an acronym often used to describe the characteristics of well-formed user stories. It stands for:

Independent: User stories should be self-contained and not dependent on other stories to be completed. This allows for flexibility in prioritization and sequencing of work.

Negotiable: User stories should be open to discussion and refinement. They are not detailed specifications but rather placeholders for a conversation between the development team and stakeholders.

Valuable: User stories should provide value to the end user or customer. They should focus on delivering functionality or solving a problem that is important to the user.

Estimable: User stories should be small enough to estimate accurately. They should not be too large or vague, making it difficult to estimate the effort required to implement them.

Small: User stories should be small enough to be completed within a single iteration or sprint. They should represent tangible, incremental improvements to the product.

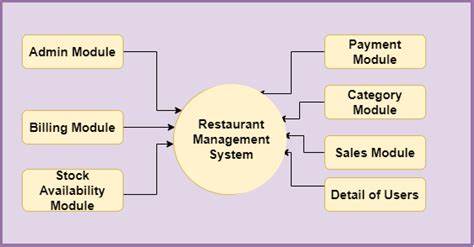
Testable: User stories should be testable, meaning there should be clear criteria for determining when the story is complete and working as expected.

By ensuring that user stories meet these criteria, teams can effectively plan, prioritize, and deliver valuable functionality to their users in an iterative and incremental manner.

Eg: User Story: "As a user, I want to filter search results by cuisine type."

* **I**ndependent: Can be implemented without relying on other stories.
* **N**egotiable: Open to discussion and refinement.
* **V**aluable: Provides value by improving the search experience.
* **E**stimable: Effort can be estimated accurately.
* **S**mall: Focused and achievable within a single iteration.
* **T**estable: Criteria for acceptance can be clearly defined.

**Restaurant management System:**



1. User Module:
   * As a user, I want to create an account so that I can access the restaurant management system.
   * As a user, I want to log in to my account to access personalized features and information.
   * As a user, I want to update my profile information, such as my name, email, and contact details.
   * As a user, I want to view my order history to track past transactions.
   * As a user, I want to reset my password if I forget it.
2. Admin Module:
   * As an admin, I want to log in to the admin panel to access management features.
   * As an admin, I want to add, edit, or delete restaurant listings in the system.
   * As an admin, I want to manage user accounts, including creating, editing, or deactivating them.
   * As an admin, I want to view reports and analytics on restaurant performance, user activity, and sales.
3. Category Module:
   * As a user, I want to browse restaurants by category (e.g., Italian, Mexican, Chinese) to find the type of cuisine I'm craving.
   * As an admin, I want to create, edit, or delete restaurant categories to organize listings effectively.
   * As a user, I want to filter search results by category to narrow down my options.
4. Billing Module:
   * As a user, I want to view my current order and its total cost before proceeding to checkout.
   * As a user, I want to add or remove items from my order before finalizing it.
   * As a user, I want to apply discounts or promotional codes to my order if available.
   * As a user, I want to view a breakdown of charges, including taxes and fees, before making payment.
5. Payment Module:
   * As a user, I want to choose from multiple payment methods (e.g., credit card, PayPal) to complete my order.
   * As a user, I want to securely enter my payment details during checkout.
   * As a user, I want to receive a confirmation email or message after successful payment.
   * As an admin, I want to view payment transaction logs and track payment statuses for orders.
6. **Sprint Backlog**:
   * A list of tasks the team plans to complete during a sprint.
7. **Sprint**:
   * A short period (usually 1-4 weeks) for the team to complete a set of tasks.
8. **Product Owner**:
   * The person who decides what features the team should work on next.
9. **Product Backlog**:
   * A list of all the features and tasks that need to be done for the product.
10. **Scrum Master**:
    * The person who helps the team follow the Scrum process and removes any obstacles they face.
11. **Increment**:
    * A small, usable improvement to the product made during a sprint.
12. **Daily Scrum**:
    * A short meeting each day where the team discusses what they did yesterday, what they plan to do today, and any problems they're facing.
13. **Sprint Retrospective**:
    * A meeting at the end of each sprint where the team discusses what went well, what could be improved, and how to make those improvements.

**How is detail added to user stories?**

Detail can be added to user stories in two ways:

* By splitting a user story into multiple, smaller user stories.
* By adding [conditions of satisfaction](https://www.mountaingoatsoftware.com/blog/clarifying-the-relationship-between-definition-of-done-and-conditions-of-sa) (acceptance criteria).

When a relatively large story is split into multiple, smaller agile user stories, it is natural to assume that detail has been added. After all, more has been written.

