**User Stories:**

* A user story is the smallest unit of work in an agile framework.
* A user story is an informal, general explanation of a software feature written from the perspective of the end user or customer.
* They are short, simple descriptions of a feature or requirement told from the perspective of the end-user.
* Describes the user or person who will be interacting with the feature. This could be a specific user type or a broader category of users.
* user stories are a versatile tool that helps Agile teams effectively capture, communicate, and deliver value to users throughout the software development lifecycle.

how user stories apply to real-time applications:

* **Identifying User Needs**: User stories help teams understand the needs and expectations of end-users when interacting with real-time applications. For example, a user story for a **live chat** application might focus on enabling users to send and receive messages in real-time.
* **Communicating Requirements**: User stories provide a simple and concise way to communicate requirements between different stakeholders, including product owners, developers, testers, and designers. They serve as a common language that everyone involved in the project can understand.
* **Guiding Development**: User stories provide a clear understanding of what needs to be built and why. They serve as a **guide for development teams** during implementation, ensuring that the final product meets the intended user needs and expectations.
* **Project Initiation**: User stories can be used during project initiation to gather initial requirements and define the scope of the project.
* **Development**: User stories guide the development process by providing clear requirements and acceptance criteria for implementing features.

**EXAMPLE:**

**User Story Format: User Story Example:**

As a <user role>,(who) As a LinkedIn User

I want to <action>,(what) I want to search for jobs that are remote only,

So that <value>, (why) So that I can apply to jobs that allows me to

work from anywhere.

**Given/When/Then Format: Acceptance Criteria Example:**

**Given** - context or Pre-condition **Given** I am under the job tab in

**When** – some action is carried out LinkedIn,

**Then** – observable outcome & expected results **When** I search for jobs

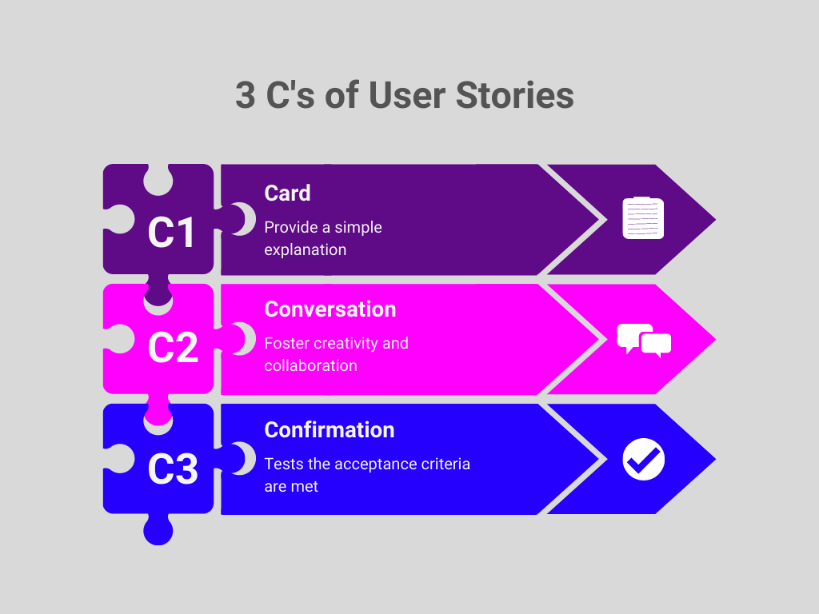
**And** I filter by remote only

**Then** remote only job are

Displayed

**And** I can apply for them

* The user expresses a need for remote jobs on LinkedIn.
* LinkedIn filters job listings based on the user's preference for remote jobs.
* The user views the filtered job listings tailored to their preference for remote work opportunities.



**INVEST :**

The INVEST criteria is a set of characteristics used to evaluate user stories in Agile development.

Each letter stands for a different attribute:

**Independent:** The user story should be self-contained and not dependent on other stories. For example, a LinkedIn user story about updating profile information should not rely on another user story about adding new profile sections.

**Negotiable:** The details of the user story should be open to discussion and negotiation between the development team and stakeholders. For instance, the specifics of how profile information is displayed on LinkedIn can be negotiated based on user feedback.

**Valuable:** The user story should deliver value to the end user. For example, a LinkedIn user story about receiving personalized job recommendations based on profile information adds value by helping users discover relevant job opportunities.

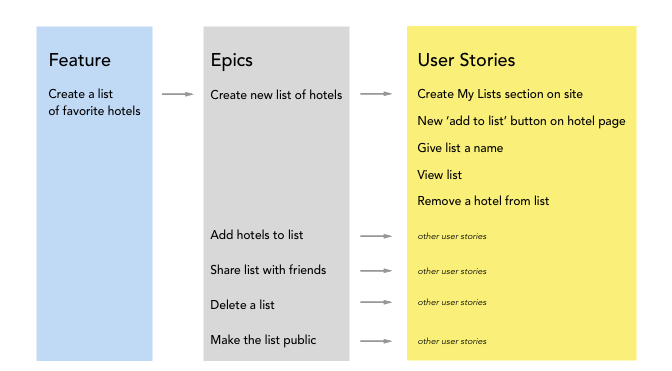
**Estimable:** The user story should be clear and understandable enough for the development team to estimate the effort required to implement it. For instance, a LinkedIn user story about implementing a feature to endorse skills on profiles should have clear acceptance criteria for developers to estimate the work involved.

**Small:** The user story should be small enough to be completed within a single iteration or sprint. For example, a LinkedIn user story about improving the search functionality within job listings should be small enough to be implemented and tested within a sprint timeframe.

**Testable:** The user story should have clear acceptance criteria that define when the story is considered complete and working as expected. For example, a LinkedIn user story about enabling users to upload a resume when applying for jobs should have testable criteria related to successful resume uploads and error handling.

**EPICS:** an Epic is a large body of work that can be broken down into smaller, more manageable user stories.

* It is a piece of work that’s broken down into specific tasks based on the needs of customers or end-users.



**Scrum: Scrum:** Scrum is an agile framework that includes several key features designed to facilitate iterative and incremental development, collaboration, and adaptability.

main features of Scrum:

1. **Iterative and Incremental Development:** Scrum emphasizes delivering workable increments of the product in short iterations called Sprints. Each Sprint typically lasts 2-4 weeks and results in a potentially shippable product increment.
2. **Roles:** Scrum defines three primary roles:
   * **Product Owner:** Represents the stakeholders, prioritizes the Product Backlog, and ensures the team delivers value.
   * **Scrum Master:** Facilitates the Scrum process, removes impediments, and helps the team adhere to Scrum principles and practices.
   * **Development Team:** Self-organizing, cross-functional team responsible for delivering the product increment in each Sprint.
3. **Artifacts:**
   * **Product Backlog:** A prioritized list of all desired features, enhancements, and bug fixes maintained by the Product Owner.
   * **Sprint Backlog:** Subset of the Product Backlog items selected for implementation in the current Sprint.
   * **Increment:** Potentially shippable product increment created during the Sprint, representing the sum of all completed and tested work.
4. **Events (Ceremonies):**
   * **Sprint Planning:** Collaborative meeting where the Product Owner and Development Team select items from the Product Backlog for the upcoming Sprint and create a Sprint Goal.
   * **Daily Stand-up (Scrum):** Short daily meeting where team members synchronize their work, discuss progress, and identify any obstacles.
   * **Sprint Review:** Meeting at the end of the Sprint where the team demonstrates completed work to stakeholders and collects feedback.
   * **Sprint Retrospective:** Meeting at the end of the Sprint where the team reflects on their processes and identifies opportunities for improvement.
5. **Empirical Process Control:** Scrum is based on the principles of transparency, inspection, and adaptation. It emphasizes empirical process control, allowing teams to continuously inspect and adapt their processes and deliverables based on feedback and changing requirements.
6. **Cross-functional Teams:** Scrum teams are self-organizing and cross-functional, meaning they have all the skills necessary to deliver a potentially shippable product increment. This encourages collaboration and collective ownership of the work.
7. **Time-Boxed Sprints:** Sprints have fixed durations and are time-boxed, typically lasting 2-4 weeks. This time constraint encourages focus, predictability, and regular delivery of value.
8. **Definition of Done (DoD):** Scrum teams define a clear and agreed-upon Definition of Done for each user story or product increment. The DoD outlines the criteria that must be met for work to be considered complete and potentially shippable.

