**UIT University**

**Department of Engineering and Technology**

**SET 212 Software Requirement and Design**

Lab#2

**Lab Title:**

1. Writing Effective User Stories

**Name of Student: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Roll No: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Sec. \_\_\_\_\_\_\_\_\_\_\_**

**Date of Experiment: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Marks Obtained/Remarks: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Lab Objective:**

To understand the structure and purpose of user stories, and to practice writing user stories for different stakeholders in a project.

**Overview**

User stories are short, simple descriptions of a feature told from the perspective of the person who desires the new capability, usually a user or customer of the system. This lab will guide you through the process of writing effective user stories for various stakeholders in a project.

**Material Needed:**

* Whiteboard or flip chart
* Markers
* Sticky notes
* Access to a project management tool (e.g., Jira, Trello, or Asana)
* User Story template handout (optional)

**THEORY**

**Summary:*A user story is an informal, general explanation of a software feature written from the perspective of the end user. Its purpose is to articulate how a software feature will provide value to the customer.***

It's tempting to think that user stories are, simply put, software system requirements. But they're not.

A key component of agile software development is putting people first, and a user story puts end users at the center of the conversation. These stories use non-technical language to provide context for the development team and their efforts. After reading a user story, the team knows why they are building, what they're building, and what value it creates.

User stories are one of the core components of an agile program. They help provide a user-focused framework for daily work — which drives collaboration, creativity, and a better product overall.

**What are agile user stories?**

A user story is the smallest unit of work in an agile framework. It’s an end goal, not a feature, expressed from the software user’s perspective.

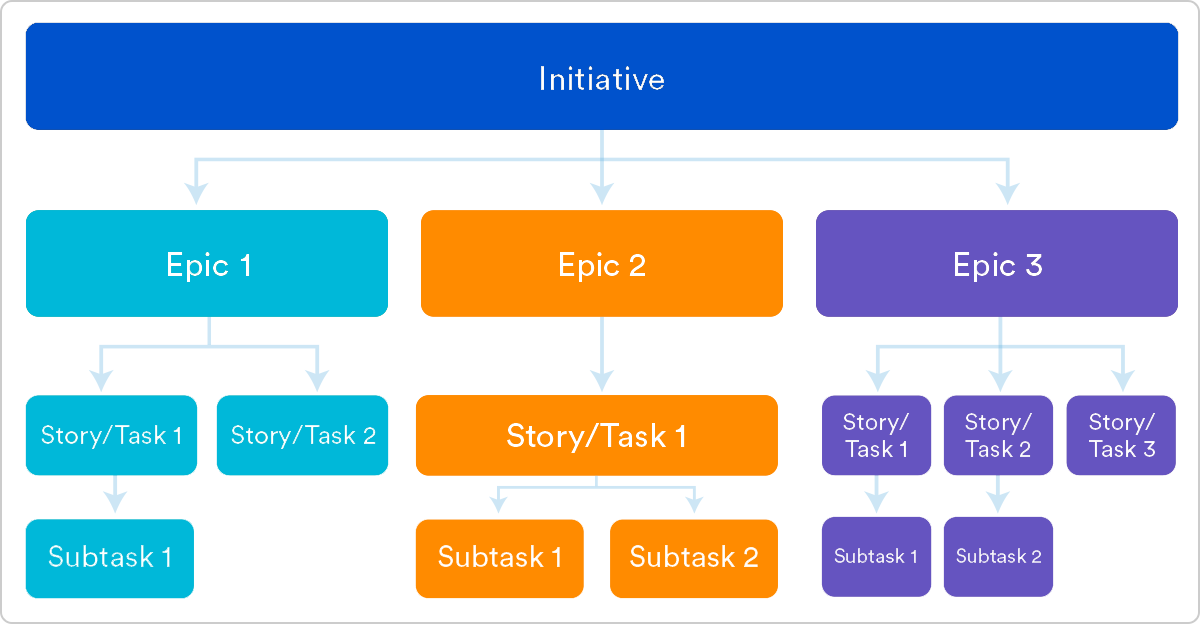
A user story is an informal, general explanation of a software feature written from the perspective of the end user or customer.

The purpose of a user story is to articulate how a piece of work will deliver a particular value back to the customer. Note that "customers" don't have to be external end users in the traditional sense, they can also be internal customers or colleagues within your organization who depend on your team.

User stories are a few sentences in simple language that outline the desired outcome. They don't go into detail. Requirements are added later, once agreed upon by the team.

Stories fit neatly into agile frameworks like [scrum](https://www.atlassian.com/agile/scrum) and [kanban](https://www.atlassian.com/agile/kanban). In scrum, user stories are added to sprints and “burned down” over the duration of the sprint. Kanban teams pull user stories into their backlog and run them through their workflow. It’s this work on user stories that help scrum teams get better at [estimation](https://www.atlassian.com/agile/project-management/estimation) and sprint planning, leading to more accurate forecasting and greater agility. Thanks to stories, kanban teams learn how to manage work-in-progress (WIP) and can further refine their workflows.

User stories are also the building blocks of larger agile frameworks like epics and initiatives. Epics are large work items broken down into a set of stories, and multiple epics comprise an initiative. These larger structures ensure that the day-to-day work of the development team (on stores) contributes to the organizational goals built into epics and initiatives.



Why create user stories?

For development teams new to agile, user stories sometimes seem like an added step. Why not just break the big project into a series of steps and get on with it? But stories give the team important context and associate tasks with the value those tasks bring.

User stories serve a number of key benefits:

* **Stories keep the focus on the user.** A to-do list keeps the team focused on tasks that need to be checked off, but a collection of stories keeps the team focused on solving problems for real users.
* **Stories enable collaboration.** With the end goal defined, the team can work together to decide how best to serve the user and meet that goal.
* **Stories drive creative solutions.** Stories encourage the team to think critically and creatively about how to best solve for an end goal.
* **Stories create momentum.** With each passing story, the development team enjoys a small challenge and a small win, driving momentum.

**Working with user stories**

Once a story has been written, it’s time to integrate it into your workflow. Generally, a story is written by the product owner, product manager, or program manager and submitted for review.

During a sprint or iteration planning meeting, the team decides what stories they’ll tackle that sprint. Teams now discuss the requirements and functionality that each user story requires. This is an opportunity to get technical and creative in the team’s implementation of the story. Once agreed upon, these requirements are added to the story.

Another common step in this meeting is to score the stories based on their complexity or time to completion. Teams use t-shirt sizes, the Fibonacci sequence, or planning poker to make proper estimations. A story should be sized to complete in one sprint, so as the team specs each story, they make sure to break up stories that will go over that completion horizon.

**How to write user stories**

Consider the following when writing user stories:

* **Definition of “done”** — The story is generally “done” when the user can complete the outlined task, but make sure to define what that is.
* **Outline subtasks or tasks** — Decide which specific steps need to be completed and who is responsible for each of them.
* **User personas** — For whom? If there are multiple end users, consider making multiple stories.
* **Ordered Steps** — Write a story for each step in a larger process.
* **Listen to feedback** — Talk to your users and capture the problem or need in their words. No need to guess at stories when you can source them from your customers.
* **Time** — Time is a touchy subject. Many development teams avoid discussions of time altogether, relying instead on their estimation frameworks. Since stories should be completable in one sprint, stories that might take weeks or months to complete should be broken up into smaller stories or should be considered their own epic.

Once the user stories are clearly defined, make sure they are visible for the entire team.

**User story template and examples**

User stories are often expressed in a simple sentence, structured as follows:

**“As a [persona], I [want to], [so that].”**

Breaking this down:

* "As a [persona]": Who are we building this for? We’re not just after a job title, we’re after the persona of the person. Max. Our team should have a shared understanding of who Max is. We’ve hopefully interviewed plenty of Max’s. We understand how that person works, how they think and what they feel. We have empathy for Max.
* “Wants to”: Here we’re describing their intent — not the features they use. What is it they’re actually trying to achieve? This statement should be implementation free — if you’re describing any part of the UI and not what the user goal is you're missing the point.
* “So that”: how does their immediate desire to do something this fit into their bigger picture? What’s the overall benefit they’re trying to achieve? What is the big problem that needs solving?

For example, user stories might look like:

* As Max, I want to invite my friends, so we can enjoy this service together.
* As Sascha, I want to organize my work, so I can feel more in control.
* As a manager, I want to be able to understand my colleagues progress, so I can better report our success and failures.

This structure is not required, but it is helpful for defining done. When that persona can capture their desired value, then the story is complete. We encourage teams to define their own structure, and then to stick to it.

**Getting started with agile user stories**

User stories describe the why and the what behind the day-to-day work of development team members, often expressed as persona + need + purpose. Understanding their role as the source of truth for what your team is delivering, but also why, is key to a smooth process.

Start by evaluating the next, or most pressing, large project (e.g. an epic). Break it down into smaller user stories, and work with the development team for refinement. Once your stories are out in the wild where the whole team can see them, you’re ready to get to work.

Effective user stories are essential in Agile development for ensuring that the team understands and delivers what users need. Here’s a breakdown of the key characteristics:

1. **Independent**: User stories should be self-contained, meaning they can be developed and delivered independently of other stories. This allows for flexibility in prioritization and helps prevent bottlenecks.
2. **Negotiable**: User stories are not set in stone; they should be open to discussion and negotiation. This encourages collaboration between stakeholders and developers, allowing for adjustments based on feedback and changing requirements.
3. **Valuable**: Each user story must deliver value to the end user or customer. If a story doesn’t provide value, it may not be worth implementing, ensuring that the team focuses on the most impactful features.
4. **Estimable**: A user story should be clear enough that the development team can estimate the effort required to complete it. If a story is too vague or complex, it’s difficult to gauge how much work is involved.
5. **Small**: User stories should be small enough to be completed within a single iteration or sprint. This makes them easier to manage and helps teams deliver incremental value quickly.
6. **Testable**: Effective user stories must have clear acceptance criteria, allowing the team to verify that the story has been completed successfully. This ensures that the delivered functionality meets the intended requirements

**Hypothetical Project: Online Meal Delivery App**

**Step 2: Identify Stakeholders**

**1. Group Discussion: List of Stakeholders**

For the Online Meal Delivery App project, the following stakeholders could be involved:

* **End Users**: Customers using the app to order meals.
* **Chefs/Restaurants**: Partner establishments providing the meals.
* **Admins**: Staff managing the app’s operations and customer support.
* **Product Owners**: Individuals responsible for the vision and roadmap of the app.
* **Developers**: Technical team building and maintaining the app.
* **Marketing Team**: Responsible for promoting the app and acquiring users.
* **Delivery Drivers**: Individuals responsible for delivering meals to customers.
* **Investors**: Stakeholders funding the project and interested in its success.

**2. Select Stakeholders for Lab Activity**

For the lab activity, we can choose the following three stakeholders:

1. **End Users**: They provide insights into user needs and preferences, which are critical for designing the app's features and user interface.
2. **Chefs/Restaurants**: Their perspective is essential for understanding menu options, pricing, and the logistics of meal preparation and delivery.
3. **Developers**: They will share insights on technical feasibility, development timelines, and potential challenges in implementing the app's features.

This selection allows for a balanced view of user experience, operational needs, and technical considerations, leading to a well-rounded discussion and planning session.

**Step 3: Writing User Stories for the Online Meal Delivery App**

**1. Divide into Groups**

Participants can be split into three groups, each focusing on one of the selected stakeholders:

* **Group 1**: End Users
* **Group 2**: Chefs/Restaurants
* **Group 3**: Developers

**2. Draft User Stories**

Here are sample user stories for each group based on their assigned stakeholder:

**Group 1: End Users**

1. **As an end user, I want to easily browse the menu so that I can quickly find meals that I want to order.**
2. **As an end user, I want to save my favorite meals so that I can reorder them easily in the future.**
3. **As an end user, I want to track my delivery in real-time so that I know when to expect my meal.**
4. **As an end user, I want to leave reviews for meals so that I can share my experiences with others.**
5. **As an end user, I want to receive notifications about special offers and discounts so that I can save money on my orders.**

**Group 2: Chefs/Restaurants**

1. **As a restaurant partner, I want to update my menu items easily so that I can reflect changes in availability or pricing.**
2. **As a restaurant partner, I want to receive notifications when a new order is placed so that I can prepare it promptly.**
3. **As a restaurant partner, I want to view sales analytics so that I can understand which meals are popular and adjust my offerings accordingly.**
4. **As a restaurant partner, I want to manage delivery times and slots so that I can optimize my kitchen workflow.**
5. **As a restaurant partner, I want to communicate directly with customers regarding order specifics so that I can ensure satisfaction.**

**Group 3: Developers**

1. **As a developer, I want to implement a secure payment gateway so that users can make transactions safely.**
2. **As a developer, I want to ensure that the app is responsive so that it works well on both mobile and desktop devices.**
3. **As a developer, I want to create an admin dashboard for managing users, orders, and restaurants, so that the admin can easily oversee the platform.**
4. **As a developer, I want to set up a robust API for integrating third-party services like delivery tracking, so that we can enhance user experience.**
5. **As a developer, I want to build automated testing for the app so that we can maintain high quality and catch bugs early.**

**3. Use Sticky Notes**

Have each group write their user stories on sticky notes, making sure to use clear, concise language. This allows for easy movement and visibility during the next steps of prioritization and discussion. Each group can then present their user stories to the whole group for feedback and refinement.

**Use online collaboration tools**

Trello: A card-based project management tool where you can create lists for different stakeholders and add user stories as cards. It’s great for tracking progress and prioritization.

**Group Activity:**

* Present user stories
* Explain the rational behind each user story
* Refine User Stories

**Individual Activity Tasks:**

* Summarize key learnings about user stories.
* Ask participants to write 5 user stories for a project they are currently working on or familiar with, applying the principles learned during the lab.
* Discuss how the concepts can be applied in future projects.

**Assessment:**

* Participation in group discussions and presentations.
* Quality of user stories written, based on criteria discussed.
* Engagement during feedback sessions.

**Resource:**

* <https://www.atlassian.com/agile/project-management/user-stories>
* <https://www.atlassian.com/agile/scrum>
* <https://www.atlassian.com/agile/kanban>
* <https://trello.com/guide/trello-101>