COSC2299 - Software Engineering Process and Tools

Tutorial 2

Pre-Requirement

Have access to Git repository – should have received this in the previous week.

Theory Recap for this Tutorial: Epics, Stories and Tasks

In Agile, epics represent high-level, large-scale pieces of work, user stories represent specific user requirements, and tasks represent the actionable steps needed to implement the user stories or epics. The hierarchy from epic to story to task helps teams break down and manage the complexity of a project in an Agile environment.

Epic:

An epic is the highest level of granularity in Agile project management. It represents a large body of work that is too big to be completed within a single iteration (sprint) and often spans multiple sprints or even the entire project.

- Epics are used to group related user stories together under a common theme or objective. They provide a high-level view of major features or functionalities that need to be developed.
- Epics are often created during the initial stages of project planning and serve as placeholders to prioritize and strategize the development efforts.
- Examples of epics in an Uber-like application could be "Real-time Location Tracking," "Payment Processing," or "User Authentication."

User Story (Story):

A user story is a smaller, more detailed unit of work compared to an epic. It represents a specific user requirement or functionality from the perspective of an end-user.

- Each user story follows a specific format: "As a [type of user], I want [an action or feature] so that [benefit or value]."
- User stories are typically small enough to be completed within a single iteration (sprint), providing a tangible goal for the development team to work on.
- They serve as a basis for planning, estimation, and testing, and are used to capture the specific needs and expectations of the end-users.
- Examples of user stories in an Uber-like application could be "As a rider, I want to enter my pickup location and destination so I can quickly request a ride," or "As a driver, I want to see the estimated fare for a ride before accepting so that I can make an educated decision."

Task:

A task is the lowest level of granularity in Agile project management. It represents a specific unit of work that is needed to implement a user story or an epic.

- Tasks are actionable items that break down user stories or epics into smaller, manageable steps.

- They are created during sprint planning and represent the actual work that the development team needs to perform.
- Tasks are usually more technical and detailed than user stories, focusing on specific development, testing, or design activities.
- Examples of tasks in an Uber-like application could be "Set up endpoint for user registration," "Implement mobile app location tracking,"

PRACTICAL

Mock up Project

This tutorial is based on a made-up project that you will find at the bottom of this document. Use it to practice your project planning and management skills.

Client Meeting

As for every Week, you will meet with your tutor to discuss the progress with your group project. As there were no group tasks last week, focus on introducing the group and how you have complementary skills. Keep the meeting brief and clean. Prepare ahead of time what you are going to be saying. Let your lead start the introduction and lead the conversation.

For this tutorial, you can find all the templates in the Scrum resources in Canvas

Group Task 1- Stories Definition (20mins)

Your main objective is to transform the following requirement into Stories and Tasks. For simplicity we will keep the feature itself as an Epic.

> *"User Registration and Profile Management Epic: As a student, I want to create an account using my university email and set up a comprehensive profile that includes my academic interests, preferred study locations, and study times, so that I can find compatible study buddies and connect with them easily."*

Discuss with your group and discuss the high-level stories for this specific Epic.

Use the following as an example

User Story 1: **As** a new user, **I want** to be able to create a new account with my university email address, **so that** I can join the Study Buddy Finding App and start looking for study buddies.

There are many more stories which are functional requirements.

Prepare a document and list as many stories as you can using the template provided in the resources.

Group Task 2 - Acceptance Criteria (20 mins)

Acceptance criteria typically answer the question: *"How will we know when this feature is working as expected?"*

Key characteristics of acceptance criteria:

- **Specific and Clear**: Acceptance criteria should be written in clear and straightforward language, avoiding ambiguity or confusion.
- **Measurable**: Criteria should be quantifiable, allowing easy evaluation of whether they have been met or not.
- **Testable**: Each criterion should be testable, meaning it can be verified through testing or demonstrations.
- **Related to User Stories**: Acceptance criteria are closely tied to user stories or requirements, outlining the desired outcomes of each story.
- **Consistent with Stakeholder Expectations**: The criteria must align with the expectations of the stakeholders, including the client or end-users.
- **Realistic and Feasible**: Criteria should be achievable within the project's constraints, including resources, time, and budget.
- **Focused on Deliverables**: Acceptance criteria concentrate on the final product or feature, rather than detailing the development process.

Group Task

Discuss with your group and start defining the acceptance criteria for your User Stories and append them to your previous document. You will not be able to complete all the user stories today, try to achieve your best.

This is an example using User Story 1

User Story 1:
As a new user, I want to be able to create a new account with my university email address, so that I can join the Study Buddy Finding App and start looking for study buddies.
Acceptance Criteria:
Scenario 1: Successful Account Creation

Given I am in the registration page of the Study Buddy Finding App
And enter a valid university email address
And a strong password
When I submit the form
Then the system creates an account

There are many more acceptance criteria here, having one person working on this 100% is quite normal.

Tasks Definition (20 mins)

As a group, now work on creating the tasks for the story.

Use the following as an example.

User Story 1: As a new user, I want to be able to create a new account with my university email address, so that I can join the Study Buddy Finding App and start looking for study buddies.

- UI Design:
  - Design the layout of the homepage to include a prominent search bar.
- Front-End Implementation:
  - Implement the homepage using HTML and CSS.

- Search Functionality - Front-End:
  - Set up JavaScript functions to handle user input and trigger search queries.
- Search Functionality - Back-End:
  - Design API endpoints for handling product searches.
- Database Integration:
  - Design the database schema to store product information.

Group Work

Expand the list provided in the example and append it to your document.

Group Task 3 - Tasks Estimation (20 mins)

Any given task should be small enough to be atomic. For estimating, look at playing poker estimates, this is a very standard way of estimating time
https://youtu.be/gE7srp2BzoM

Task estimation is an art that can make or break a project. At this stage, stay within the limits of your experience and follow some basic rules such as

- Do not create tasks that are too long, optimally you should be able to clearly say where you are at in your Scrum.
- Work with the team and make sure they are comfortable with the estimation.
- Have you planned all the pre requirements? Such as training
- Are there any foreseeable blockers? Firewalls you didn't know existed.

Industry relevant: As a rule of thumb, if you believe a task takes 1 hour, double it. If you are working with a client, you may want to double it again (4hrs). Remember, when you work with clients "delivering ahead of time, is good, delivering late, is horrible."

Group Task

Do a time estimation based on the work you have done today, create a burnout chart.
Example "Implement the search bar on the homepage using HTML and CSS." I would personally put 3 hours. But your estimate needs to be well connected to the delivery. Are you including JS implementation and integration as well?
This is why tasks need to be extremely well defined.

Group Task 4 – Definition of Done (20 mins)

Definition of Done typically answer the question: "How will we know when this feature done?"

Definition of Done is a team's checklist of all the criteria to be met so that a deliverable can be considered ready for customer use. It is a "generic checklist" for all User stories. Criteria that are like conditions of satisfaction that are applied across all user stories on the product backlog.

Examples of items in your Definition of Done could include:

- All testing completed
- No known defects
- Code review completed and passed
- Architecture Document updated

Group Task

Discuss with your group and start defining your definition of done and append them to your previous document.

Project for Week 3 only: Study Buddy Finding App

Project Description: Study Buddy Finding App

1. Introduction: In this university group project, our team will be developing a Study Buddy Finding Application for a client who seeks to enhance collaboration and academic success among students. The application aims to connect like-minded students who are looking for study partners to foster a conducive and supportive learning environment. The Study Buddy Finding App will be designed to run on both iOS and Android platforms, ensuring widespread accessibility to students across various devices.

2. Project Goals: The main objectives of the project are as follows:

- Develop an intuitive and user-friendly mobile application that facilitates the process of finding study buddies.
- Enable students to create profiles and specify their study preferences, subjects, and availability.
- Implement a matching algorithm to connect compatible study buddies based on common interests and schedules.
- Provide real-time messaging functionality to allow students to communicate and coordinate study sessions effectively.
- Create a secure environment that protects user privacy and personal information.
- Incorporate features for users to rate their study sessions and provide feedback, helping to improve the overall experience.

3. Features and Functionality: The Study Buddy Finding App will encompass the following key features:

- User Registration and Profile Creation: Students can sign up using their university email addresses and create detailed profiles, including academic interests, preferred study locations, and study times.
- Study Buddy Matching: The application will employ a sophisticated algorithm to match students with compatible study buddies based on their shared interests, courses, and availability.
- Real-time Messaging: Users will have access to a messaging system that enables seamless communication with their study buddies to plan and coordinate study sessions effectively.
- Study Group Creation: Students will have the option to form study groups, allowing multiple individuals with similar study preferences to collaborate on specific academic subjects or projects.
- Ratings and Feedback: Users can rate their study buddies and leave feedback after each study session, fostering a sense of accountability and continuous improvement within the study buddy community.
- Account Settings: Users will be able to manage their account settings, including profile information, notification preferences, and study preferences.

4. Project Scope: The scope of the project encompasses the design, development, and testing of the Study Buddy Finding App. The application will prioritize functionality and user experience to ensure a seamless and reliable platform for students to connect and study together.

5. Project Timeline: The project will be executed over the course of one semester, with the following key milestones:

- Project Planning and Requirement Gathering: Weeks 1-2
- UI/UX Design and Prototyping: Weeks 3-4
- Frontend and Backend Development: Weeks 5-10
- Testing and Debugging: Weeks 11-12
- Finalization and Client Presentation: Weeks 13-14

6. Collaboration and Responsibilities: The team will collaborate closely throughout the project, with each member assigned specific roles and responsibilities based on their expertise. Regular meetings and progress updates will be held to ensure effective communication and coordination.

7. Conclusion: By the end of the project, we aim to deliver a fully functional Study Buddy Finding App that meets the client's requirements and enhances the study experience for university students. The application's success will be measured by its usability, positive user feedback, and the number of successful study partnerships formed through the platform. With a well-designed and efficient app, we look forward to fostering a supportive academic community and helping students achieve their academic goals collaboratively.