Agile Manifesto Exercise

Case Study: Creating an Agile Manifesto with a Student Project

Background

A group of students from a software engineering course has been tasked to create a digital Agile Manifesto as part of their semester project. The purpose of this project is to simulate the Agile software development process, allowing students to understand and practice Agile principles, roles, and ceremonies. The students will work in a team and apply Agile methodologies such as Epic, Features/Stories, Sprints, Story Points, and Subtasks to complete their project.

Project Objective

To develop a web-based Agile Manifesto platform where users can:

1. Learn the values and principles of Agile.
2. Access interactive examples and resources.
3. Collaborate and share ideas about Agile methodologies.

Step-by-Step Execution

1. Project Kickoff

The instructor acts as the Product Owner and provides the following requirements:

- The platform should have a homepage that explains the Agile Manifesto.
- Users should be able to explore Agile values and principles interactively.
- A resource library for Agile practices (Scrum, Kanban, etc.) must be included.
- Collaboration features like a discussion forum or comments section should be implemented.

The team holds a kickoff meeting to:

- Understand the project goals.
- Assign roles (Scrum Master, Developers, Tester).
- Define the project scope.

2. Create the Epic

The team defines the Epic for the project:

Epic: "Create a web-based platform to educate and engage users about the Agile Manifesto and methodologies."

3. Break Down into Features and Stories

The team breaks the Epic into smaller Features and Stories:

Feature 1: Homepage for Agile Manifesto

- **Story 1.1:** Design and implement a visually appealing homepage.
- **Story 1.2:** Add an introduction to Agile and its history.
- **Story 1.3:** Include navigation to other sections of the platform.

Feature 2: Agile Values and Principles

- **Story 2.1:** Create a page to display the four core Agile values.
- **Story 2.2:** Implement an interactive feature to explain the 12 principles.
- **Story 2.3:** Add examples to illustrate each value and principle.

Feature 3: Resource Library

- **Story 3.1:** Design a library for Agile practices (Scrum, Kanban, etc.).
- **Story 3.2:** Include downloadable resources like templates and guides.
- **Story 3.3:** Add search and filter functionality.

Feature 4: Collaboration Features

- **Story 4.1:** Develop a discussion forum for users.
- **Story 4.2:** Enable users to comment on resources and principles.
- **Story 4.3:** Implement user authentication for collaboration features.

4. Sprint Planning

The project is divided into four 2-week Sprints:

- **Sprint 1:** Complete the Homepage for Agile Manifesto.
- **Sprint 2:** Build the Agile Values and Principles section.
- **Sprint 3:** Develop the Resource Library.
- **Sprint 4:** Implement Collaboration Features and perform final testing.

The team estimates Story Points for each story using Planning Poker:

- Small stories: 1-2 points.
- Medium stories: 3-5 points.
- Large stories: 8-13 points.

5. Subtasks for Stories

The team breaks each story into Subtasks. For example:

Story 1.1: Design and implement a visually appealing homepage

- **Subtask 1.1.1:** Create wireframes for the homepage.
- **Subtask 1.1.2:** Develop HTML and CSS for the layout.
- **Subtask 1.1.3:** Integrate animations using JavaScript.
- **Subtask 1.1.4:** Conduct user testing and gather feedback.

Story 4.2: Enable users to comment on resources and principles

- **Subtask 4.2.1:** Design the comment input field and UI.
- **Subtask 4.2.2:** Develop backend APIs for storing comments.
- **Subtask 4.2.3:** Implement frontend integration with APIs.
- **Subtask 4.2.4:** Test the comment functionality.

6. Daily Stand-ups

Every day, the team holds a 15-minute stand-up meeting to discuss:

- What was completed yesterday.
- What will be worked on today.
- Any blockers or issues.

7. Sprint Reviews

At the end of each Sprint, the team presents their progress to the Product Owner (instructor). Feedback is gathered and incorporated into the next Sprint.

8. Sprint Retrospectives

After each Sprint, the team holds a retrospective to discuss:

- What went well.
- What could be improved.

- Actionable steps for the next Sprint.

9. Final Delivery

At the end of the fourth Sprint, the team delivers the final project. The platform includes:

- A polished homepage.
- Interactive pages for Agile values and principles.
- A resource library with search functionality.
- A discussion forum with user authentication.

Lessons Learned

Through this project, the students learned:

- How to break down a complex project into manageable parts using Epics, Features, and Stories.
- The importance of estimating work with Story Points.
- How to collaborate effectively using Agile ceremonies.
- How iterative development leads to continuous improvement.
- The value of receiving and incorporating feedback from stakeholders.