Method Selection and Planning (2)

Group Name:
Cohort 1, Group 4

Group Number:
Cohort 1, Group 4

Members:
Jude Hall
Rosie Hogg
Ishraan Ismail
Sam Wildgust
Ruby Hanson
Tom Devany
Tomas Asllani

Introduction

This document is an updated version of the Assessment 1 “Method Selection and Planning” deliverable inherited from the previous team. It has been revised to reflect the requirements of Assessment 2, where development focused on extending and integrating with an existing codebase, which is what the brownfield development is based on. Updates clarify the software engineering methods used, strengthen the project plan approach, and better align planning activities with testing, integration, and continuous development practices adopted by our team.

Method Selection

As a team, we have adopted the Agile Methodology due to its support for iterative development and flexibility when working with complex systems. In Assessment 2, development focused on extending and integrating with an existing codebase rather than building a system from scratch. Agile development allows software to be developed in small stages, with each iteration improving on the previous one while reducing the risk of breaking the existing functionality. Furthermore, using an agile approach also allows us to deliver software quickly to our clients, ensuring that we meet their requirements and enable them to see their product frequently.

Since Agile is our overall philosophy, we made use of the Extreme Programming (XP) and Scrum frameworks during the development process. We chose to use selected XP practices because it provides practical engineering practices such as incremental implementation, frequent testing, and continuous integration, which were particularly important when modifying an existing system. Additionally, XP enables us to maintain a simple implementation, frequent releases and iterations. It is much more efficient to work on one feature at a time and have another team member test these while others are being developed. XP promotes these practices and prevents us from developing the entire project before we discover issues with the implementation.

Using Scrum-inspired practices for project management provides a flexible way to apply agile principles. We can organise our work into sprints, keeping the project manageable as team members can focus on one small feature at a time. Additionally, scrums enable better team collaboration, where everyone knows what they're working on and how their tasks fit into the sprint goal.

Tools

Documentation

For our documentation, we chose to use Google Docs because it is free to use and allows for real-time collaboration, enabling multiple team members to work on the same document simultaneously. Furthermore, with all files stored on the cloud, team members can access documentation anytime, anywhere, and from any device. Additionally, Google Docs automatically saves all changes, eliminating the risk of losing work. Likewise, the version history feature enables us to easily track edits and restore previous versions if needed.

Microsoft Word was considered as an alternative way of documentation due to its ease of use and familiarity with all team members. However, we chose Google Docs instead due to its real-time collaboration capabilities, which are essential for an Agile team, as we can see updates instantly and leave comments for discussion. Also, the auto-save feature is crucial for preventing the risk of losing work, which would waste time and reduce productivity. This toolset was retained in Assessment 2 to maintain continuity with the documentations.

Communication

As a team, we chose WhatsApp as our primary communication method because all group members were already familiar with the app and had it installed. Furthermore, using a mobile app enabled us to communicate quickly, reducing delays compared to alternative methods such as email. The app also works on tablets and computers, allowing everyone to stay connected wherever they are.

Version Control

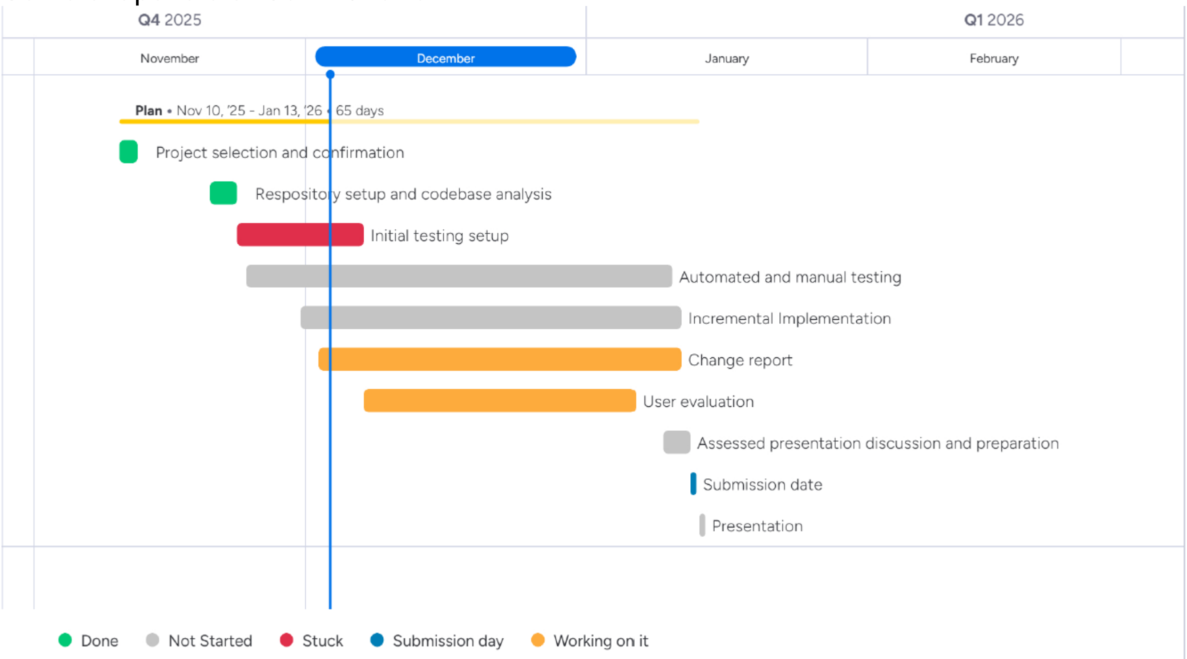
We chose GitHub as our version control system because it supports code collaboration through the use of branches. These branches allow multiple people to work on the same project simultaneously, which is essential for our agile approach. Branches will enable us to do this without accidentally overwriting someone else's work. Changes can then be merged smoothly using pull requests. In addition, GitHub maintains a full history of changes to code and documentation, allowing us to review and restore previous versions if necessary. GitHub was also integrated with continuous integration tools, allowing automated builds and tests to run when changes were pushed or merged, helping detect integration issues early.

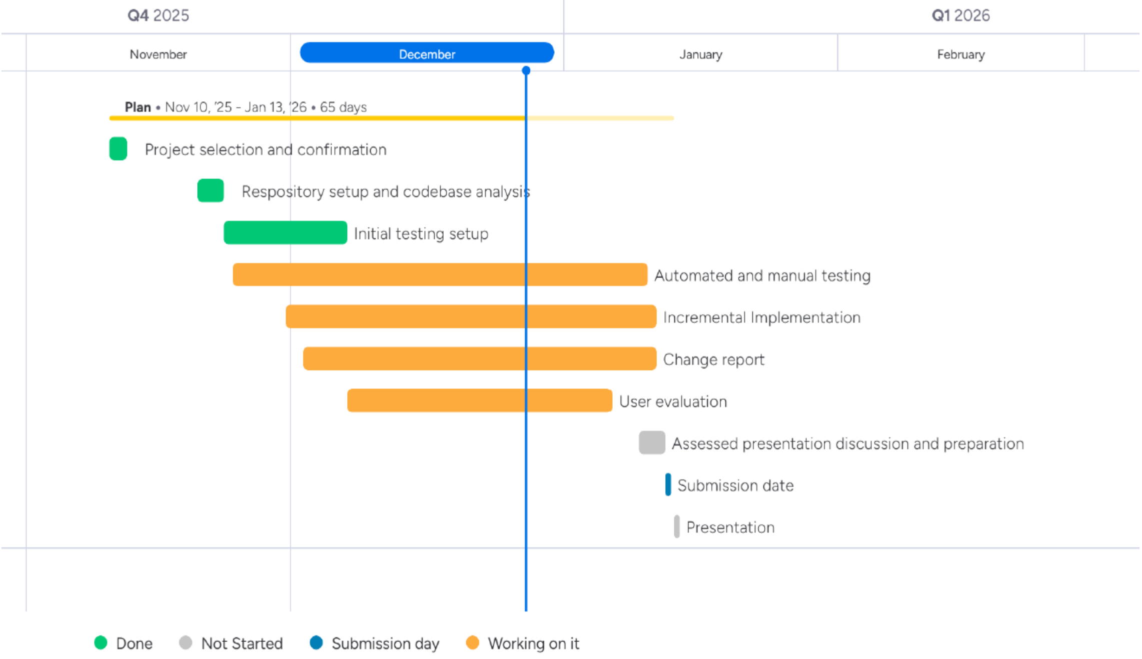
Organisational Approach

At the start of the project, we listed everyone's strengths so we knew who could do what best and could assign tasks accordingly. We decided this was the best approach because it builds teamwork and understanding, reduces confusion later, and helps the project flow more smoothly. Furthermore, it also highlighted gaps in our skills, which we can develop later on by taking on tasks that weren't our initial strengths.

Every week, we meet in person to review our progress on the project and set deadlines for the upcoming week. A Gantt chart is created, and updated so we can visually see the tasks to be completed. This approach allows all team members to see what needs to be done and by when. Improving team coordination and making it easier to track our progress. Furthermore, it keeps the team accountable and helps us identify when a team member is struggling to meet deadlines, so we can take appropriate steps to address it. We assign multiple people to a task to increase the speed and efficiency with which we complete our set deadlines. Moreover, it improves accuracy and quality by having multiple people review the same work, ensuring our project is of the highest quality. During our meetings, we also use that time to make decisions about our project as a group. We chose this approach over a virtual one because we found it easier to communicate our ideas face-to-face than over a call. In addition, we can ask questions, clarify points, and reach agreements more quickly than during an online call.

Some snapshots of Gantt Charts:





Project Plan

Assessment 2 began with the selection of an existing project developed by another team. The selection process involved taking notes of different groups during the presentations and by evaluating the group’s deliverables for assessment 1. Based on this review, a project was selected and confirmed on 12th November, after which development activities for assessment 2 began.

Following project selection, the inherited GitHub repository was cloned and made private in accordance with assessment 2 requirements. Initial effort was dedicated to understanding the existing codebase, system architecture and requirements, before any functional changes were made. This analysis informed the planning decision. Continuous integration was established early by configuring automated build verification and dependency graphs, ensuring that new changes were validated before being merged into the main code base.

Testing began early in the project. Initial challenges were encountered with JUnit, particularly when setting up the testing environment for the inherited codebase. Initial testing focused on checking that the existing functionality worked as expected. As development progressed and new events and features were added, the extent of testing was gradually increased. Throughout the development, manual and automated tests were being conducted.

During week 11, user evaluation activities were conducted to gather feedback on gameplay, usability and overall experience. Participants were provided informed consent through signed ethical approval forms. Feedback collected was used to identify areas for improvement.

Throughout assessment 2, documentation was continuously being developed and updated to reflect ongoing development and changes. Regular team discussions through WhatsApp ensured that the development progress, testing, documentations and other assessment requirements remained aligned.

In the later stages of the project, attention shifted towards consolidation and review. Development activity focused on integration stability, addressing issues identified through testing, and preparing for the assessed presentation conducted in Week 13.