Method Selection and Planning

Team 2 - Billy’s Amazing Team

Member Names: Mitchel Bekink, Alyx Bruno-Bamford, Ashley Bryan, Rajul Chawla, Bosco Lau, Kacey Van Der Walt, Chris Grzywacz

Software Engineering Methods

Outline:
Our approach to this project was most closely aligned with the waterfall methodology. We devised our requirements and used them to inform our design/architecture which then allowed for an implementation that reflects this process.

This is justified as we were able to concretely identify what the user was looking for in this project. The waterfall approach suits projects where user requirements are static, so an iterative process is not required. This is due to the fact that this project had only a singular meeting with the user, and as such the lack of client feedback to interim designs would lead to the project being unsuitable for the other design we considered, the agile methodology.

We did deviate slightly from the waterfall model when it came to writing up our formal documentation, which was done in parallel to the implementation instead of in series. This is because it was not part of the software designing process itself; it was just writing up the earlier design plans in a more formal manner.

Collaborative Tools:

- Whatsapp: This was used to schedule meetings, inform the group about absences and as a quick way for team members to provide links to software to the group. All members of the team were already familiar with it and found it easy to use. It was a better solution than a text group chat as it was available on both phones and laptops, so we could still communicate with a team member if they had a faulty device, minimising risk. Emails would also satisfy that requirement but were not used as it was easier to read logs of past messages on WhatsApp.

- Zoom: This was used for online meetings. It was chosen over another service discussed by the team, Discord, as it was easier to set up. It also had an integrated whiteboard which the team used for prototypes of the system architecture which was not present in Discord and it would have required an additional third-party program to achieve the same result.

- Google Workspace (drive, docs, sheets and jamboard): These were used to write up and store most of both the planning and final documents for the project as well as a collection of useful information the team wished to store. It was chosen over Microsoft Office as it would allow for simultaneous editing of the same document by multiple team members no matter what device they were using. It would also make it simpler to access the documents in question as if the office were used, we would need to transfer documents to each other through. Google Drive handles this automatically.

- GitHub: Used for version control. This was used to allow multiple team members to work on the software's implementation simultaneously. There was limited discussion of alternatives to Github as many members of the team were already familiar with it before the project had begun.

- PlantText: Used to create long-term schedules. This particular tool was a quick and easy way to make Gantt charts, which were used to organise deadlines. Part of the reason this was used was due to its ability to create UML diagrams for software architecture planning. This cut down on the amount of programs the team had to familiarise themselves with over the project.

Approach to team organisation

Leadership and Decision Making:
Initially, our team decided on a dispersed structure, where all group members generally had an equal say in decisions. We did not assign specialised leadership roles, instead we opted to split the work amongst group members. As the project developed, we shifted from a work style where all members would work together on each deliverable, to one where we split the deliverables amongst ourselves so 2 - 4 people would work on each deliverable. This altered the leadership dynamic, as we agreed that the people working on each deliverable would have a higher say on how that deliverable would be finalised.

This approach was appropriate as we worked well together without a designated leader. In some team environments, a leader would be beneficial to ensure the work is being managed effectively and to make decisions when needed. However, for our team, we were happy to use a majority system for decisions generally and to allow the groups working on deliverables to make the decisions on their deliverables.

In addition, it was also appropriate for this project due to the small size. The first assessment has a time period of only 6 weeks, and a relatively small number of deliverables given our team size. So this allowed us the flexibility to bring every issue up with the whole team, as in much larger projects with more deliverables, involving the whole team in differing decisions would be impractical.

Work Distribution:
The process we used to distribute the work required for each deliverable is similar to our leadership system. Everyone was allowed to tell the team what deliverables they wished to work on for various reasons such as enjoying the type of work involved or wanting to build their skills in a certain area.

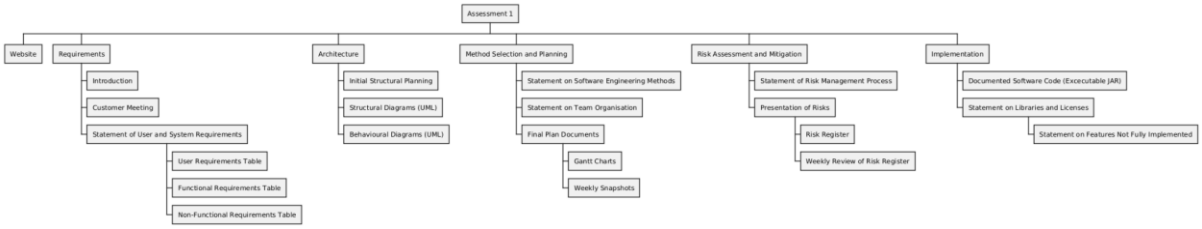
This system worked well, as everyone wanted to work on different areas of the project, so the only tricky part was ensuring that the correct number of people were working on each deliverable. As deliverables are worth different marks and some require more work than others, we decided each deliverable would have at least 2 people working on it. For small deliverables like the website and team organisation: we assigned one team member to create it, and another member to review it. This ensured that if one team member was unable to work on it, there would be someone else able to take over.

This organisation approach was appropriate for our team as we all naturally had differing opinions on the work we each wanted to do. This allowed the work distribution process to run smoothly without conflict. If we had many people wanting to work on one deliverable and others that were neglected, we would have had to implement a different approach that could handle these conflicts.

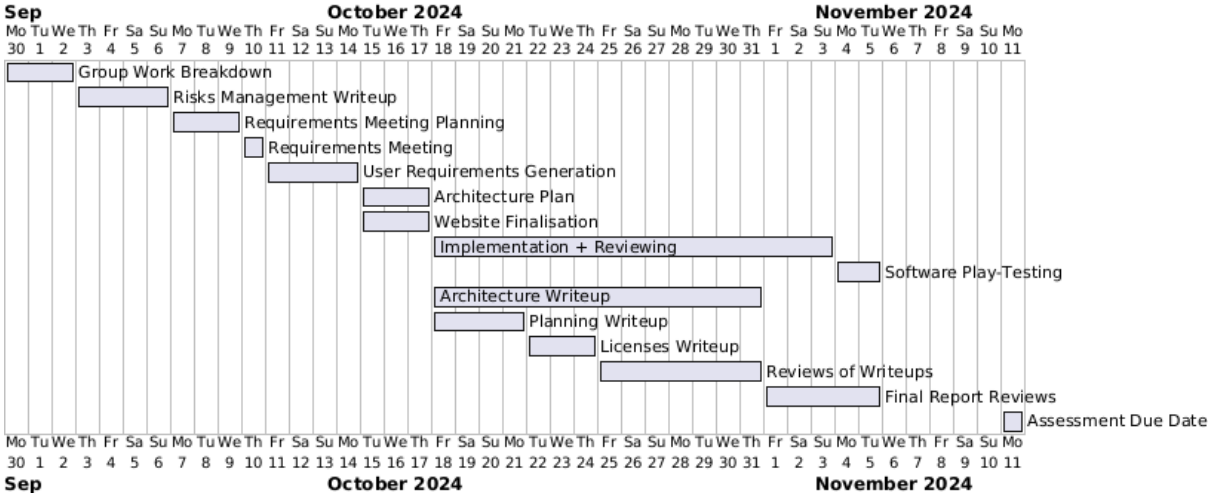
This approach was appropriate for the project due to the small project size and small team size. The small project size meant that we could assign work to members as a team, whereas if there were more deliverables, a more rigid work assignment approach would have been needed.

Systematic Plan

The first step we took in planning our project was to create a work breakdown diagram. This split each project deliverable into its different sub-tasks, allowing us to better understand what was required of us in this project. It also allowed us to condense the assessment document into an easily viewable document. Our work breakdown diagram is shown below:
Link to work breakdown diagram: [4.1 Work breakdown diagram]



The main foundation of our plans throughout the project is our Gantt charts. During week 2 of our project, as a team we created our initial Gantt chart, breaking up the timeline of our project into manageable tasks, each with its due date. Dependencies were managed by ensuring that tasks that had dependencies were scheduled after the tasks that they depended on. For instance, the Implementation task was scheduled to start as soon as the requirements gathering was done. We decided to give all tasks equivalent priority, as different team members would be assigned different tasks, so tasks could be worked on in parallel rather than one at a time where possible. Our initial Gantt chart is shown below:
Link to initial Gantt chart: [4.2 Week 1 Gantt chart (30-09)]



The dependencies above are limited, with the only dependencies being:

- ‘Software Play-Testing’ being dependent on ‘Implementation + Reviewing’
- ‘Implementation + Reviewing’ being dependent on ‘Architecture Plan’
- ‘Architecture Writeup’ being dependent on ‘Architecture Plan’
- ‘Architecture Plan’ being dependent on ‘User Requirements Generation’

Our plan evolved over time due to various reasons, such as some tasks taking longer than anticipated and team member illness. We met and updated the plan on a weekly basis to stay on top of these factors. On the following page is a week-by-week breakdown of our plans' evolution, each with a link to the corresponding Gantt chart on our website.
Week 1:

This was our initial Gantt chart based on the work breakdown diagram. We assigned tasks to different team members as follows:

- Group Work Breakdown, Requirements Meeting Planning, Requirements Meeting, User Requirements Generation, Architecture Plan: Entire team
- Risks Management Writeup: Mitchel, Kacey, Rajul and Chris
- Website Finalisation: Yawshi
- Implementation + Reviewing: Alyx and Ashley
- Architecture Writeup: Kacey and Mitchel
- Planning Writeup: Mitchel
- Licences Writeup: Ashley
- Reviews of Write-ups: Rajul and Chris
- Short tasks were done by various members, based on who was available at the time

Link to Gantt Chart (30-09): 4.2 Week 1 Gantt chart (30-09)

Week 2:
No obvious changes this week, all was going according to the original plan.
Link to Gantt Chart (7-10): 4.3 Week 2 Gantt chart (7-10)

Week 3a:
No obvious changes at the beginning of this week, all going according to the original plan.
Link to Gantt Chart (14-10): 4.4 Week 3a Gantt chart (14-10)

Week 3b:
The plan had quite a large change later in the week. The architecture planning took a lot longer than originally planned, so we had to push back the entire project schedule by a few days to allow extra time for this, as almost the entire project was dependent on the architecture. We also decided to extend the time available for software play-testing, to allow us to iron out any issues that may arise during this stage.
Link to Gantt Chart (19-10): 4.5 Week 3b Gantt chart (19-10)

Week 4:
No changes this week, all going according to the original plan after last week's alterations.
Link to Gantt Chart (21-10): 4.6 Week 4 Gantt chart (21-10)

Week 5:
Only a minor change this week, we decided to have all writeups and reviews complete for the 7th of November, this way we could have everything done for our in-person meeting on Thursday morning. It would also give us a bit more time in case there is anything wrong with the project at this point.
Link to Gantt Chart (28-10): 4.7 Week 5 Gantt chart (28-10)

Week 6:
Another small change this week. We have had to extend the allotted time for the implementation and architecture writeup to be done before the 7th as these took us a little longer to complete than originally planned. This means that all the play-testing and final report reviews will be done in person at our Thursday meeting.
Link to Gantt Chart (04-11): 4.8 Week 6 Gantt chart (04-11)

Bibliography

[1] WhatsApp, “WhatsApp,” WhatsApp, 2024. [Online]. Available: https://www.whatsapp.com/.

[2] Zoom, “Zoom,” Zoom Video Communications, 2024. [Online]. Available: https://www.zoom.com/.

[3] Google, “Google Workspace,” Google, 2024. [Online]. Available: https://workspace.google.com/.

[4] GitHub, “GitHub,” GitHub, 2024. [Online]. Available: https://github.com/.