Method Selection & Planning

Group 18

Team B

Olivia Betts
Zac Bhumgara
Nursyarmila Ahmad Shukri
Cameron Duncan-Johal
Muaz Waqas
Oliver Northwood
Teddy Seddon

Our team uses Agile software development as our software engineering method. Short development cycles are used in the agile project management technique, which prioritises continuous improvement in the creation of products and services. Requirements and solutions are developed in collaboration amongst self-organising cross-functional teams.
We chose this to ensure we were constantly checking the code we had produced against the requirements.
Other development methodologies considered but ultimately discarded were the waterfall method (outdated and not useful for changing requirements) and spiral (ideal for large, risky projects - neither of which this is).

One of the collaboration tools that we have used is Github. It is used to store our code files as it makes it easier for collaborating on our software project. It allows multiple people to work on the same codebase at once, as well as cloud storage, and the ability to manage code collisions

We also used Google Docs to write our deliverables, again to allow multiple people to edit the same document. These were stored in a shared Google Drive, so that they could be accessed easily and from multiple devices. Another bonus of using Google Docs was that the files can easily be downloaded to .pdf format, which was the required format for the deliverables.
We considered using OneDrive, however it was more effort to set up, and more annoying to ensure everyone had access; so we didn't use it in the end.

Other than that, we have created a group email so that it is easy to send an email message to everyone at once and also to ensure that everyone receives any new updates through email.

The way we have approached team organisation has been by focusing on fairness, efficiency and through addressing any potential bottleneck issues.

The focus in the team organisation is on equality regarding workload. We tried to split the project as evenly as possible (~12.8 marks per person) to encourage an equal contribution from each team member rather than heavy lifting from some and not others. This will keep team morale high, since no one will feel injustice with their workload. For example, if one team member was designated 5 marks and another 20 marks, then the one who is assigned significantly more will feel annoyance towards the other creating a bad work environment. So, by balancing the marks, work production is allowed to be as efficient as possible.

We have also considered the possibilities that some members might fail to complete their assigned work. This could involve a minor setback, perhaps a team member being ill, or a much more impacting setback with a member dropping out of university. Because of this we will ensure that there is not a bus factor of 1 for each section, meaning there are at least 2 people working on each deliverable. This will allow work to continue even if someone fails to contribute, meaning the project can still be completed. This risk has been further mitigated by the designation of significantly more people for the more critical tasks, for example the implementation and architecture deliverables have 4 people working on them since the workload required is much higher than the others.

We will also communicate regularly with one another via discord to see where people are at with their work. This will be regulated by an assigned leader who will check that each section is being worked on and that everyone is carrying out their role. This hierarchical structure will be enhanced with the idea of sub-leaders who are assigned the role of leader for their particular deliverable. Their job will be to communicate with the team leader if their members are doing their tasks, making decision making much easier and further guaranteeing the project’s completion. The role of leadership is essential in ensuring that the flow of work and communication is maintained throughout the course of the project, this not only keeps the team morale high but also speeds up the work rate.

Finally, the use of weekly Gantt charts throughout the project will be used to track each team’s progress. This establishes a point of reference for each deliverable within the project so we can assess who’s done what and when to see if each team member has provided sufficient input.

Task	Start date	End date	Dependencies
Requirements Elicitation	16/11/22	30/11/22	
Initial Architecture	23/11/22	30/11/22	Requirements elicited (doesn't have to be perfectly finalised)
Risk Assessment	7/12/22	14/12/22	Risks must be identified before writeup can begin
Implementation	18/1/23	31/1/23	Requirements, Architecture
Architecture evaluation	25/1/23	31/1/23	Implementation, Initial architecture

We have also made a Gantt chart laying this out visually, which can be found on our website TeamBEng1.github.io