

Method Selection and Planning

ENG1 Team 16, 'Team Team'

Luke Batten

Owen Crucifix

Samuel Humphreys

Robbie Parr

Jude Daniels-Smith

Alan Yang

Methodology

For this project, we will be using an agile software development methodology, SCRUM. We decided to go with SCRUM because of its adaptability, customer involvement, focus on teamwork and time. Since the project can change frequently with regard to requirements, the versatility of this method is ideal. As requirements change, we can cycle back around to make changes all the way to the implementation of the game. This keeps the client involved, as we are always able to ask for feedback in order to make changes. Finally, within SCRUM there are sprints that break down the big problems into smaller more manageable parts. The SCRUM master is also key as it helps with management and ensuring everyone is doing the work needed

Tools for collaboration

As a team, we use several tools and programs. For communication between our weekly meetings, we are using Discord. Discord allowed everyone to access it from both their computer and their phones, meaning quick access from anywhere. Alternatives would have been WhatsApp or Snapchat. Discord allowed groups to be made more easily, for example, one for each part of the project. Different areas are discussed in different sections, so information is not lost and easy to locate. Also, discord allows for images to be shared with ease

For the team's weekly meeting, we used discord again. Alternatives would have been Microsoft teams or zoom. We felt that discord was easiest since we were already using it for team communication also, we did not need to keep signing into different applications.

For our customer meeting, we use Zoom as our client said they would prefer to use this over other options, for example, meeting in person. Zoom meant more ability but also the team was not at risk of getting covid.

Google Drive is a tool we use for working on our documentation. It allows us to share documents amongst ourselves and for team members to simultaneously work together on them. This was the only tool we considered using because everyone in our team had a lot of experience using it.

We use GitHub to share code and put all the final documents there to be shown on the website. Other tools are Visual Studio Live Share and code together, but everyone had some experience with GitHub which also allowed for the easy creation of the website as well.

For game development, we chose libGDX since it is cross-platform, a requirement gathered from the customer in our requirements meeting. Additionally, libGDX uses Java, which was the required programming language for the project.

Team organisation

Firstly, Jude Daniels-Smith took on the role of SCRUM master, being responsible for breaking up larger problems into more manageable parts. As a team, we discussed our individual skills, assigning more tasks to where their strengths lie. Luke and Jude stated they are strong on the documentation side, so took the lead on requirements, method selection and planning and risk assessment. Owen was keen to help with the architecture of the implementation - the classes for which are to be discussed further below. Robbie started his prior experience in web design, so he took the lead on the website. Finally, Wynn and Alan, with assistance from the rest of the team, took control of the implementation since they had previous skills working with graphic libraries. Over the whole task, Robbie was extremely helpful with making sure everyone was keeping up with their tasks.

Breakdown - who did what

Website- Robbie

Requirements- Jude and Luke

Architecture – Owen

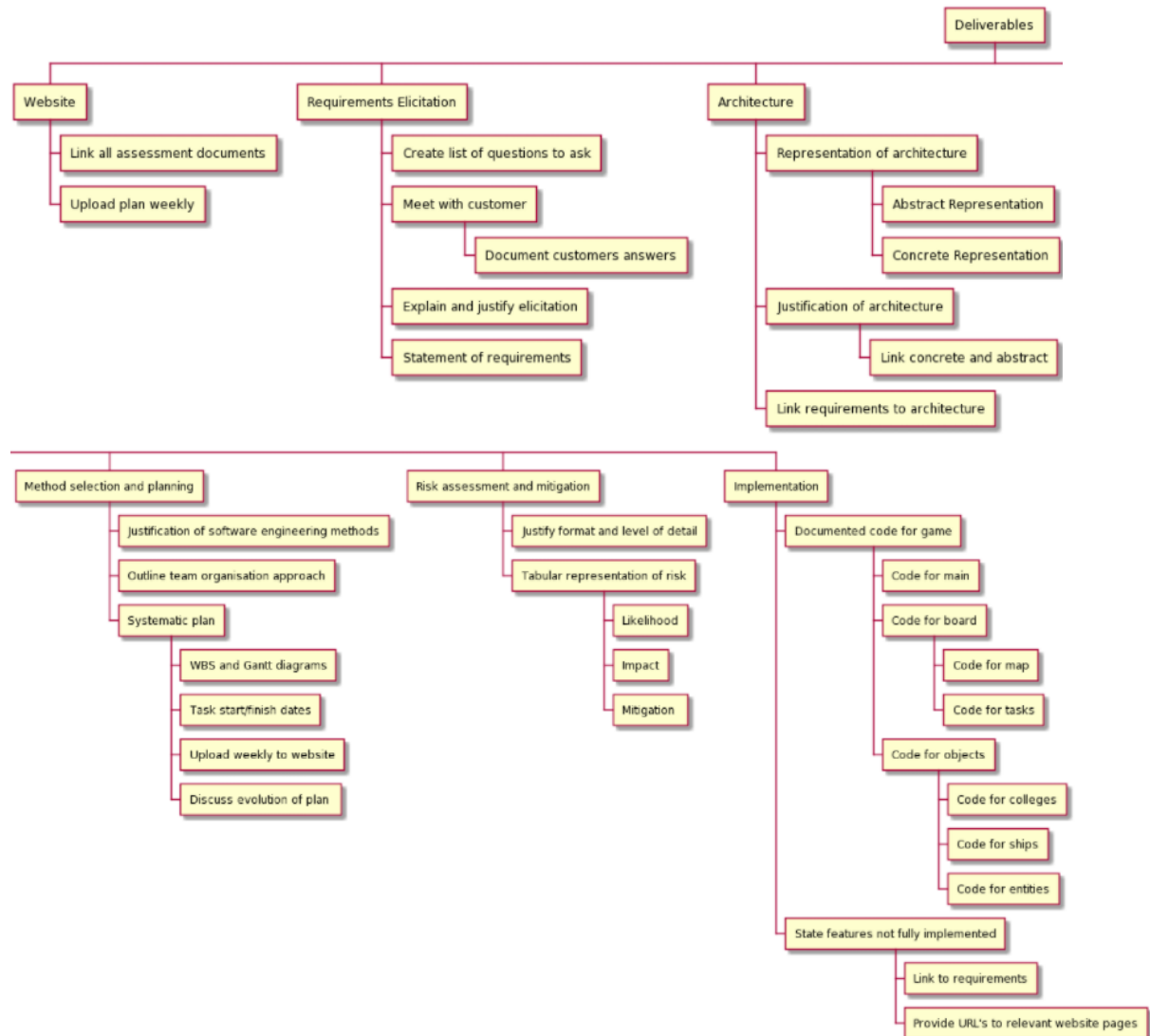
Method selection and planning – Jude and Luke

Risk assessment and mitigation – Jude and Luke and Robbie

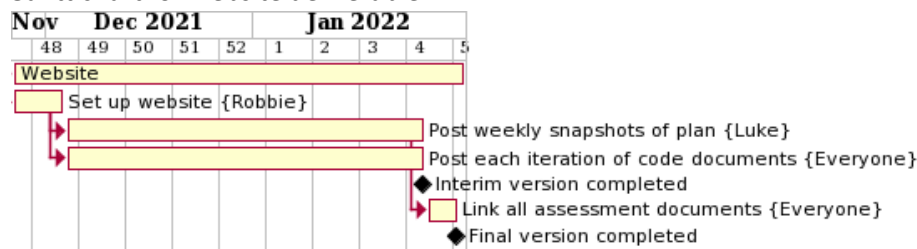
Implementation - Wynn and Alan took the lead.

Project Schedule:

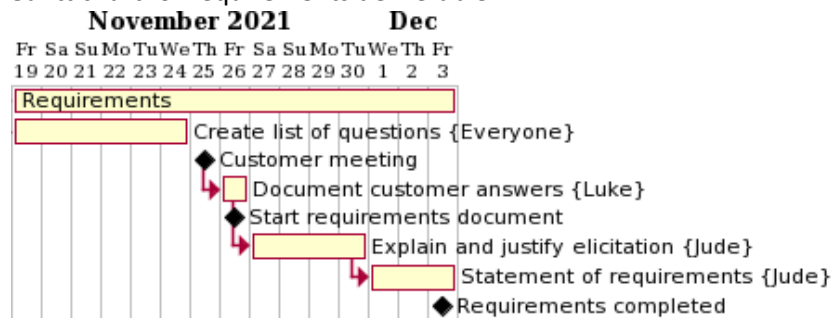
WBS describing the deliverables of the project, as well as the tasks contained within them.



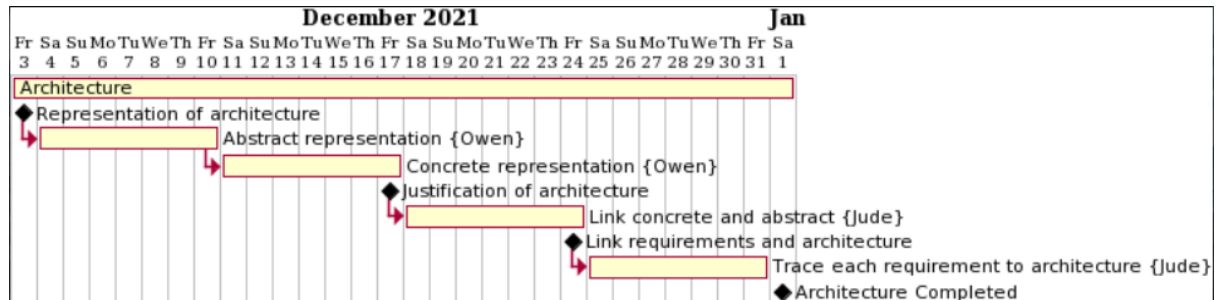
Gantt chart for website deliverable:



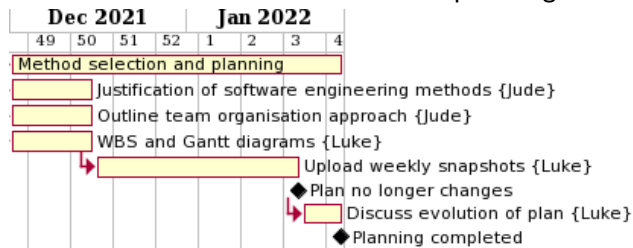
Gantt chart for requirements deliverable:



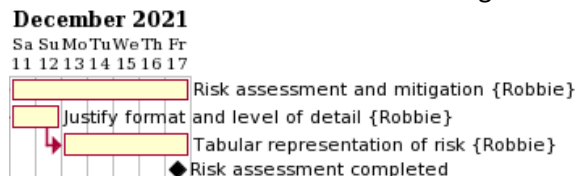
Gantt chart for architecture deliverable:



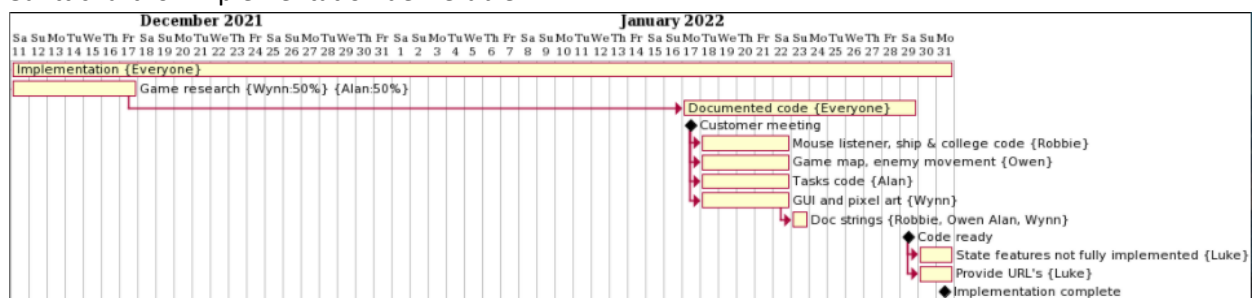
Gantt chart for method selection and planning deliverable:



Gantt chart for risk assessment and mitigation deliverable:



Gantt chart for implementation deliverable:



Evolution of project plan:

As discussed above, we had to decide our development approach and elicit requirements to begin our project plan. We held a meeting early in the project to discuss ideas of what we thought the game could be like, which were put together to form a list of questions to ask in our initial customer meeting in order to elicit requirements. Once we knew our approach and what was expected of us we discussed the areas of the project that we were most confident in tackling. Once the above had been established we were able to begin the completion of deliverables for the project.

Our choice of the SCRUM approach and weekly online team meetings via discord meant that progress was very easy to track. We would agree upon tasks to complete either individually or as small subgroups for that week, which helped to break down the project. Since we had biweekly meetings (One in-person and one online) and setbacks/changes in the scenario could be easily accounted for. This meant the specifics of the plan changed plenty of times, as tasks could be transferred between members if necessary, in turn providing a safe bus factor. Finally, the plan changed often as we realised that completing the deliverables sequentially was not always essential or even beneficial.