

# Method Selection And Planning 1

Group 2 - Vikingz

Damian Boruch

Tommy Burholt

Elliott Bryce

James Butterfield

Ren Herring

Zhenggao Zhang

Sharlotte Koren

# Method Selection and Planning

## Software Engineering Methods

### Programming Methodology

When working on a large project it is essential to use an appropriate programming methodology to keep the project organised and delivered on time. We researched 3 different programming methodologies and compared their strengths and weaknesses.

We first looked into the waterfall methodology, this is where you follow a linear development path going from plan to final product. It is divided into different phases of development such as Requirements and Implementation which needs to be reviewed and signed off before moving onto the next phase. Although this is the simplest to understand and use, it doesn't promote change because once you have signed off on a phase you can't go back and change it. For this reason we didn't choose to use the waterfall methodology as we will constantly be suggesting changes and new ideas to improve the project.

Following this, we researched the spiral methodology. In this methodology you repeatedly work through development cycles which consist of 4 stages; Planning, Risk analysis, Engineering and Evaluation. This allows for constant iteration and improvement of the software since flaws and new ideas are always being evaluated and implemented. However, this methodology requires a lot of planning and management which is intended for larger projects alongside a focus on risk analysis which isn't needed for this project.

Finally we settled on the agile methodology since it best matched the needs of our project. You initially come up with a plan and a list of requirements to outline the scope of the project. Then you create a backlog of tasks which need to be completed in each week's sprint. A sprint consists of 5 stages; Design, Development, Testing, Deployment and Review. This allows us to have greater flexibility during the development process and allows our team to collaborate effectively on the project due to its need for constant communication between team members.

### Collaboration and Development Tools

As a team we decided to go with the google suite as our main set of tools due to their great collaboration features, ease of use, and native version control features. We explored other sets of tools such as the 'Microsoft 365' set of tools, however the university already provides us with the google suite and it would be too much of a hassle to migrate over. Following this, we decided that all of our documentation will be written in google docs, our log book will be kept in google slides, and that we would use google sheets for tracking work distribution.

For the implementation, we decided to go with Github as our code version control system and vscode as our IDE due to our team's familiarity with it and its ease of use. We did look into using the IntelliJ IDE, however we deemed it too much of a commitment to change to an

IDE that no one in the team thought had better functionality than vscode. To create our structural, behavioural and gantt chart diagrams we went with a mix of using plant UML with the vscode diagram generation extension and Goodnotes 6. For the artwork we used a mixture of Aseprite and Photoshop to create the UI and sprites owing to their versatility and plethora of features. When making the music and sound effects we used Ableton Live 12 since it is a fully featured digital audio workstation with industry standard tools and effects.

As for communication, we chose whatsapp to be our main point of contact. We also looked into using discord due to its ability of being able to share your screen whilst in a team call, however we went with whatsapp because of simplicity and ease of setup. We also believed that whatsapp would be a better option due to the fact that everyone already had it installed on their phones which meant they would be quick to respond to any important messages if needed be, which is something very important within the agile methodology; response and engagement.

## Team Organisation

### Roles and Responsibilities

When it came to team organisation we initially wrote down what each team member wanted to work on in the project and created a spreadsheet to equally distribute the work. From this we created an initial gantt chart, outlining when different parts of the project were to be started and finished. We also created a mini gantt chart each week which showed the current week's tasks alongside an idea for what was going to get done next week (All weekly gantt charts are available on the website under the planning tab).

	Website	Requirements	Architecture	Method Selection and Planning	Risk Assessment	Implementation	Total
James	0	0	5.5	0	5	5	16
Sharlotte	0	6.66666667	5.5	0	0	0	12
Tommy	1	6.66666667	0	3.33333333	0	5	16
Damian	1	0	5.5	3.33333333	0	5	15
Elliott	0	6.66666667	0	0	5	0	12
Ren	1	0	5.5	0	0	5	12
Zhenggao	0	0	0	3.33333333	0	5	8

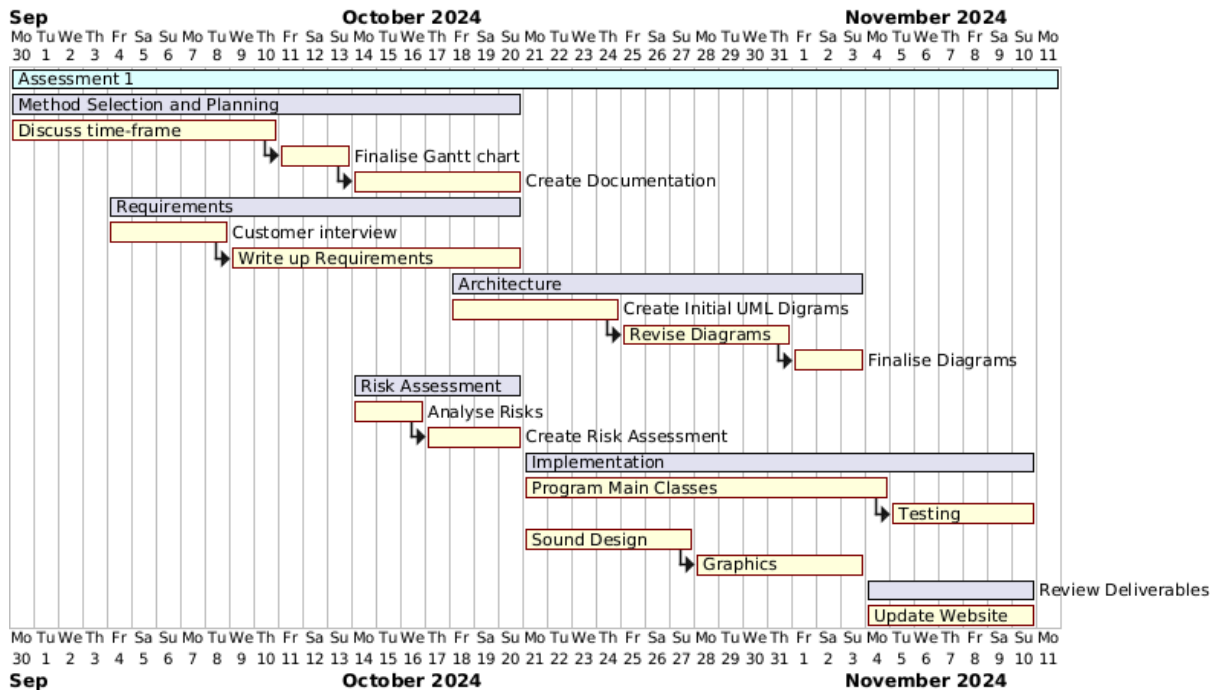
  

	Marks	Members Allocated
Website	3	3
Requirements	20	3
Architecture	22	4
Method Selection and Planning	10	3
Risk Assessment	10	2
Implementation	25	5
Peer Assessment	10	7
Total	100	

Each week we had 2 meetings, one on a Monday and the other on a Friday. On Mondays we discussed how the project was going and outlined the tasks for the week alongside creating that week's gantt chart. On Fridays we reviewed that week's work and suggested any changes which could be made over the weekend before the next meeting. In each meeting we updated our logbook which kept track of attendance and what was discussed in the meeting.

# Project Plan

## Overview



When planning the project we created an initial gantt chart, planning when work would be completed.

### Week 1 23/9/24

During week 1 we are planning on getting to know each other and each other's strengths and weaknesses. We are also hoping to talk about what tools we are going to use and how we are going to tackle the product brief as well as how we are going to share the workload.

### Week 2 30/9/24

During week 2 we are planning on having set up all of our collaboration tools that we are going to use throughout the project and start discussing who should focus on which tasks. During this week we are also planning on brainstorming questions for our client meeting which will occur on the 8th of October.

### Week 3 7/10/24

During week 3, we are planning on writing up our requirements, depending on how the client meeting goes, hopefully these should be done by the end of the week as this would allow us to slowly start thinking about architecture and implementation. We are also hoping to finalise

the weekly plan gantt charts by this point. Somewhere around this time we are also planning on putting up the website so we can track our weekly progress.

### Week 4 14/10/24

During week 4 we are planning on creating our risk assessment as well as starting work on architecture. Hopefully if we can get these things done early we will be able to start implementation a bit earlier than first hoped..

### Week 5 21/10/24

During week 5 we are hoping to be well into programming the game as this would mean we would have enough time if anything were to go wrong, or take longer than expected. We are also hoping to come to the end of our architecture document, as that would mean there would be nothing else to do but develop the game. Around this time we are also planning to start developing the assets for our game.

### Consolidation Week 28/10/24

By the end of consolidation week we want to be fully finished with all of the documentation and focus on finishing creating assets for the game and the code itself. If all of this is done we will start focusing on making sure the website is up to date.

### Week 6 4/11/24

By week 6 we are hoping to have finished everything there is to do, and will be making sure that the formatting of each deliverable is as it should be ready for submission, as well as updating the website to contain all of the relevant information in chronological order.