

Method Selection and Planning

Software Engineering Methods:

Our team have identified Scrum, a framework for Agile development, as the most appropriate method for this project. Scrum is particularly effective for small teams of between three and nine [1], as bbwe are working in a small team of six this was important to consider. Additionally, Scrum employs short iteration cycles. This involves the incremental delivery of software to the customer and fast feedback for team members. This is important to our team as we aim to have a close relationship with our customer, being guided by their demands in order to produce software that meets their needs as completely as possible. Furthermore, our team members have no experience of development on this scale and so being able to give feedback as quickly as possible is crucial in keeping everybody on the right track.

Working with short iteration cycles also affords us the ability to adapt quickly to requirement and feature changes at any stage of the development process, from our understanding of the SEPR module handling these changes will be necessary at some point. This was the main motivation for not choosing a plan-driven method for development, as plan-driven methods do not adapt well to changes. Another feature of Scrum that we feel will be useful to our team is that it is fairly easy to learn even for those with no formal experience of Agile development. Our team has no experience of Agile development and we do not want to spend too much of the planning process learning development methods.

Lastly, we feel the structure of the assessment supports the features of Scrum, from planning to implementation and testing, each stage can be broken down into simple team sprints on the go rather than having a complex planning process at the start of the project. According to last years State of Scrum report, the average sprint length amongst the surveyed Scrum users was 2.4 weeks [2]. This supports the way our team aims to work, breaking down each assessment into small blocks and regularly re-evaluating what needs to be done and how well we have completed the tasks we have set. We aim to organise weekly sprints in the early planning stages of our project and extend these sprints as we begin implementation and the flexibility of Scrum allows us to do this.

Tool Selection:

When selecting tools for development, collaboration and planning we were looking for tools that supported our working needs as well as the demands of the project. The software we were using had to be affordable, accessible on a range of platforms and devices and simple to learn.

Game Engine:

We decided to use jMonkeyEngine as the environment in which to build our game. It is a popular game engine and as such has a fairly large community. This community is useful to our team as we are new to game development and will need access to lots of learning resources. As well as being free to use, jMonkeyEngine gives us a lot of creative freedom, it supports 3D development and has a wide range of plugins available to tailor the project to

our customer's needs. Another option would be to use libGDX, which is more appropriate for 2D game development and this is another option we will explore.

Programming Language and IDE:

We have decided to use LibGDX over Jmonkey. Although Jmonkey had more documentation, it was a 3D engine and we wanted to create a 2D game. Using Jmonkey resulted in extra work in the implementation and would have caused the implementation process to be much longer. After using both we have found a much lower learning curve with LibGDX which fits better with the time constraints of the project. Using Java was part of the project brief and we all have experience programming in Java. We will use the Eclipse IDE as it can be used alongside either game engine, it is stable and completely free. We all have experience with Eclipse from our first-year modules and we are all confident working with this IDE.

Version Control and Source Code Management:

We chose to use GitHub over competing systems for a number of reasons. GitHub allows us to use the platform's tools for free as we are all students. Due to its popularity, if we encounter any problems, we will have a wealth of information to support us that might not be available with competing services such as BitBucket.

Project Management Platform:

We have decided to use Taiga as our project management platform. Taiga allows us to create a free public project with as many contributors as we need, something other platforms didn't provide. Taiga is explicitly designed for Agile development teams and it is supportive of our process. Taiga is also accessible to all members of our team due to its simple UI and integrated GitHub support. Taiga also fulfils our need for multi-device support, it works on multiple platforms and has a mobile app.

File Sharing:

We are using Google Drive to share documents. The university provides us with free Drive storage and every team member is familiar with the platform's features such as team drives from previous university projects. Google Drive also supports mobile and desktop use, a feature we will find extremely useful.

Team Communication:

Facebook Messenger is the platform we are using to communicate as a team and plan team meetings. It is convenient as it works on all devices and all members of the team already have an account and are familiar with its features. Facebook Messenger is also completely free and supports many types of communication including voice and video chat.

UML Diagram:

To produce our UML class diagram we used the ArgoUML tool. We chose ArgoUML as it was completely free, recommended on the module website and most importantly is pre-installed on the university's computers. This makes collaboration simpler for our team and also makes our diagram accessible to other module groups. Need to include UML program for concrete

Chart Production:

Google sheets was used to plan the rest of the project with a Gantt chart. This allowed easy collaboration between members allowing us all to access it at the same time. This is also easy to edit fitting our agile workflow. It was also easy for the team to use as we have all used sheets or excel before. This meant the team didn't have to learn how to use the software. Furthermore, we all have access to google apps through our university accounts.

Art and mapping tools:

We have decided to use Piskel for art creation. This is a simple tool that makes the creation of pixel art quick and simple. It has a grid allowing you to easily identify pixels that you are drawing and supports various file formats. In addition, it is a free software that is browser based, this means all of us can use the software on any computer making it very flexible. We considered alternatives such as photoshop and paint.net. However, not all of us had access to photoshop and paint.net didn't have all of the functionality we wanted. Furthermore, we are using Tiled to create tilemaps, this is designed to work with LibGDX which we are using to create our game and there were few alternatives. As a result, this appears to be the best option for creating tilemaps quickly.

Team Organisation:

Following Scrum, we agreed to have a Scrum Master we agreed that Sam would be our Scrum master, as he was our meeting chair for assessment 1 and organised assessment 1 using Taiga.io. Our Scrum Master assigns roles as the project develops, allowing a predictive and reactive approach. This fits the agile workflow as it allows us to reassess the task at regular intervals and evolve the project. This also allows work to be completed more efficiently with roles being switched when needed. This means that all team members work to their full capacity on the project.

This method is appropriate for the project as our specification is it is likely to change later in the project and we will also have to quickly adapt to working on another group's project after assessment 2. An adaptable organisation will allow us to quickly see what we need to do and assign people to the new roles we find. In addition, this will allow room for us to assign team members to new roles that appear rather than forcing a task into an old role. This will allow the team to have a better understanding of the project. As a result, the project will be of a higher quality when it is finished, however, this may cause certain tasks to take a longer time. Using a different method would require a fixed plan from the start, we currently can't do this as we don't know how the project will change and evolve. Instead, we have created an adaptable plan which will allow us to work well, but still has room for changes.

This method is also appropriate for the team as we are all equally capable of working on different tasks and this will allow us all to gain experience in these different areas. Furthermore, if we are all able to view the entire project it will allow the team to visualize the progression of the project and resolve issues that other members may not have seen. This will also keep the project interesting for all members as they will be thinking about different tasks and problems.

We have assigned some permanent roles aside from the Scrum Master. This will allow us to organise the team but does not include the software tasks which will be distributed as the project progresses. These can be found below:

Log keeper:

Lloyd will take this role after being our secretary in the first assessment. The role requires the member to keep a log of meetings and plans of the group, this will be useful for reference throughout the project.

Report Reviewers:

Ethan and Lewis will perform this role. As Report Reviewers they will proof check our reports to make sure they are up to a good standard and make sure they are grammatically correct.

Information Collectors:

Otto and Zafir will take this role. This will require them to research and store information in a meaningful way. This information will then be used to benefit the team throughout the project.

Plan for the Rest of the Project:

Assessment 3	S/W3	S/W4	S/W5	S/W6	S/W7	Assessment 4	S/W7	S/W8	S/W9	S/W10	Easter	Su/W1	Su/W2	Su/W3	Su/W4
Website						Evaluation and Testing Report									
Assessment 3 Deliverables						Explain and Justify Evaluation and Testing									
Game Executable						State Changes to Assessment 3 Testing									
Test Plan and Results						How Meets/Doesn't Meet Requirements									
User Manual (Editable and PDF Version)															
Change Report						Implementation									
Summarise approaches to change management						Implement to Meet New Requirements									
Testing Report						How Modifications Meet Requirements									
Methods															
Plan for Assessment 4															
Implementation and Report						Project Review Report									
Implement Changes to Code						Summarise Team Management and Structure									
Comment on Changes to Code						Management and Structure Evolution									
Systematic Report of Changes						Summarise Use of Methods and Tools Used									
Justification of All Significant Changes						Account of how Methods and Tools Evolved									
Report Unimplemented Features															
						Presentation and Picking of Project									
Presentation and Picking of Project						Assessed Presentation									
						Planning									
						Presenting									
Team Meetings						Team Meetings									

Software Engineering Plan for Assessment 3:

We believe a good start to the project will be looking at the changes and method for the assessment to begin with. Getting a good approach for these will be required for the rest of the assessment. We will then begin implementation quickly as testing and documentation can't begin until we have something to test. When we have finished a good amount of implementation we can easily justify changes as it will be hard to know all unimplemented features until close to the end of the assessment. However, we will keep track of any

features we decided not to implement throughout in the assessment. The majority of website features can't be shown until the end of the assessment when they are completed so the website work will be done at the end of the assessment. Also, the website changes shouldn't take a long amount of time. We will also plan Assessment 4 close to the end of Assessment 3 when have a clearer idea of what will need to be done in assessment 4.

Bibliography:

[1] Scrum Guides, (Nov, 2017), the Scrum Guide, Available:

<https://www.scrumguides.org/scrum-guide.html>

[2]Scrum Alliance, (2017), State of Scrum, Available:

[https://www.scrumalliance.org/ScrumRedesignDEVSite/media/ScrumAllianceMedia/Files%20and%20PDFs/State%20of%20Scrum/2017-SoSR-Final-Version-\(Pages\).pdf](https://www.scrumalliance.org/ScrumRedesignDEVSite/media/ScrumAllianceMedia/Files%20and%20PDFs/State%20of%20Scrum/2017-SoSR-Final-Version-(Pages).pdf)