ENG1 - Assessment 1

Method Selection and Planning

Plan1.pdf

Group 6

Freya Goodger	sg1967
Mikolaj Wyrzykowski	mw2179
Barnaby Matthews	bm1287
Cooper Love	cl2702
Oliver Cassey	oc854
Anna Hrynyshyn	ah2886
Oliver Thompson	ot699

In this project, we are choosing to use agile methods. There are many reasons agile methods fit this project best. Firstly, agile methods produce working software frequently while adapting to changing requirements [1]. We will be producing the game with reduced requirements for assessment 1, and then adapting to another team's game with the whole requirements scope for assessment 2. These requirements will fit in much better with agile methods than plan-driven methods, as plan-driven methods do not adapt to change nearly as well. This project does not require the benefits of plan-driven methods, like their use in managing large teams and increased focus on documentation. Documentation is important for this project, especially when transferring projects between teams, but it is not worth the overhead to focus on it as much as plan-driven methods require. Our team is of a small size, so the benefit of a central plan is lost as communication is much simpler. Additionally, agile methods allow us to engage with the customer better, as we can produce a functional game for them to provide feedback on. This is particularly useful as we will get many opportunities to communicate with our customer in labs, or meetings when necessary.

We will be using scrum as our agile method of choice. The high-level nature of scrum allows us to tailor it to fit our specific needs, and therefore work most effectively [2]. This advantage over other methods such as extreme programming made selecting scrum the best choice for us.

We will have week-long scrum sprints with a meeting before and after each, where we will discuss plans for the sprint before, and progress after. Focus will be placed on changes which will need to be made for the next sprint if not enough tasks were completed. The daily scrum can be discussed between collaborators on specific tasks to denote how their work is progressing when necessary, but is not strictly required due to the collaboration tools we are using to support the project.

The two main tools utilised for collaboration on this project are Google Workspace for working on documents and GitHub as our frontend for git for version control.

Google Workspace includes Google Drive and Docs, which are both vital components. Drive hosts all of our files relating to documents, mainly Docs files for the assessment and other files including notes and links. It was chosen due to its tight integration with our already existing university emails, so everyone already could easily access and utilise it. Additionally, it provides convenient integration with Docs, which is important due to how much it is used for the project. We chose Docs for several reasons: it is available in the web browser which makes it accessible for everyone, it allows simultaneous editing of documents by multiple people, and it has a detailed version history showing all changes made to the document. This last feature is particularly useful in the case that evidence is required. We considered whether we should use git to version control our whole project, using LaTeX. However, we decided Google Docs was much simpler to use, especially due to higher experience with it in the group compared to LaTeX. For version control in the actual development of the software, git is the software we selected. Git has considerably more advantages when used for software development version control compared to if we used it for the documents; to name some: allowing pull requests, forks, cloning to the local machine to be edited with an IDE and executed, advanced merging, and continuous integration. Therefore, something like Google Workspace was not considered for this part of the project. We looked at other version control software briefly before settling on using git. Git's decentralised nature compared to other version control like SVN is beneficial when developing. This as well as more experience in the group meant using git was the logical choice for us. Git will be very useful during scrum sprints, as it allows for easy modification of the software by multiple developers.

To work with git, we are using GitHub for our central repository to allow for better management of continuous integration, documentation, tracking of issues, and pull requests. We selected GitHub, as it has many very useful features for collaboration, is simple to get started with, many group members have prior experience with it, and it provides the ability to easily configure a website using GitHub Pages. The tracking of changes on GitHub will be important during scrum sprints, as they allow people to view the daily changes in a much more concise and simple manner. Additionally, continuous integration will allow for the easy deployment of functioning software at the end of each sprint, letting the executable be accessed easily by everyone and shown to the customer.

For communication, we are using Discord primarily, with email as a backup or if otherwise more useful for specific purposes. Discord is free, easy to use and set up, cross platform, and every team member has experience using it. In case there are issues with Discord, everyone has access to email each other using our university emails. We considered using Slack instead of Discord, but the listed benefits of Discord led us to choosing it, especially considering we do not find any of the benefits of Slack to be important enough.

Our team's approach to team organisation closely aligns with agile methodologies that are widely used nowadays. In particular, for our project we have chosen to use Scrum methodology as it is well-suited for both team and the project because of its emphasis on collaboration, adaptability and iterative development.

One of the key aspects of our team organisation is fostering collaboration to utilise experience, knowledge and skills of each member to implement the game and all deliverables. From the beginning, we started building trust and respect among team members, creating an environment that ensures that everyone feels comfortable sharing any ideas and potential solutions. This approach created an atmosphere that encourages brainstorming and allows us to view the project from multiple perspectives, which is crucial in the development of our project. Thus, enhancing creativity and problem solving, we easily prepare questions for the customer interview which are a fundamental part of defining user requirements. Next approach that ensures that all team members are kept informed and involved in the development process is open communication. This is achieved through utilising Discord as our primary communication method that is familiar to all team members. This enables us to keep everyone up to date with any progress and challenges that might arise, enhancing cohesion within the team.

Another cornerstone of our team organisation is task allocation based on individual skills, preferences, and knowledge. Given the project's time constraints, this approach was a strategic decision crucial for saving time and ensuring the proper execution of each development step, leveraging the expertise and expressiveness of each team member. Rather than assigning specific roles, we adopted a flexible approach where any team member willing to work on a task and available to do so could contribute. We also monitored the distribution of tasks to ensure an equitable breakdown of work among team members. As a team, our goal is to optimise productivity and prevent burnout by ensuring that each member is engaged in tasks aligned with their strengths and interests.

The last but not least important approach is feedback. As we utilise agile methodologies for our project delivery, we adopt an iterative approach to development, making feedback from other team members is crucial. Typically, it is achieved by either informing everyone during each team meeting or, if it is urgent, by posting a message in Discord, which serves as our primary communication channel.

We aim to deliver a high quality product that meets the requirements of product brief while maximising team efficiency and satisfaction. Therefore, our team's approach to organisation aims to be well-suited to both the team dynamics and project requirements. This is achieved through qualities such as collaboration, open communication, adaptability, effective task management and feedback. Additionally, our team places a strong emphasis on continuous improvement and learning, striving to adapt to changing circumstances and refine our processes as needed.

The project follows a scrum-based approach, with weekly splints, defined key tasks, starting and finishing dates, task priorities, and dependencies. Additionally, for better visualisation of how our plan evolved, we have created Gantt charts which were updated as our project progressed. The weekly snapshots of them can be found here.

Week 1

Beginning from the first team meeting during practical sessions, we actively engaged in team forming, examined the brief of the product, and defined the potential shape of our project. Starting from the first practical session when we met each other, we began developing a team identity, which helped us to familiarise with each other, thus facilitating how our team will operate for the rest of the project. Another important aspect of our meeting was the identification of the software that we would be using during the project development. We decided to mainly focus on using Google Workspace to collaboratively work on our project deliverables whilst using GitHub as version control tool. Besides that, we ensured that all team members agreed on how we would keep in touch and discussed potential approaches to team meetings and tasks. In the case of our team, Discord became the main tool for communication that everyone is comfortable with. The next activity that we undertook was an online treasure hunt where the members made a start at looking things up and started to think about who might be responsible for each subtask and we also started thinking about tasks itself. That activity enhanced our vision of the resources that would be needed in order to complete this project. Additionally, we established the initial project structure and implemented a simple CI.

Week 2

The second sprint of this project, which spanned from 22nd to February 28th began with planning of the workload for the incoming week and defining any obstacles that may occur along the way. Luckily, at this early stage, we did not accumulate any backlog that might need to be dealt with. One of the most important aspects of the development process of any project is establishing outstanding communication with the stakeholders. That is crucial to ensure that their feedback has been heard, helps identify and mitigate risks effectively and contributes to the decision-making process.

Having examined the product brief once again, we started to identify potential questions that might arise during the implementation of the project and could be answered by the product stakeholders. We compiled them all together and discussed the availability of each team member to arrange the meeting with the stakeholders. Besides that, we started planning the project by splitting the work into smaller chunks and primarily assigned team members to certain tasks, which was supported by the desire and experience that each team member had towards the task.

Week 3

During our third sprint, which started on February 29th and finished on March 6th, we conducted an interview with a customer that provided us with valuable insights and answers to our questions. We then analysed the interview and ensured that all data was nicely stored on our team’s Google Drive. This interview analysis also served as a great addition to the brief for defining both system and user requirements.

The next crucial step that we undertook was splitting the work into smaller tasks that could be assigned to the team members. In our team organisation, we did not adhere strictly to role assignments; instead, we assigned tasks based on deadlines. Additionally, during that week, two members of the team identified risks and iteratively kept monitoring and updating them over the next two weeks.

From the implementation part of the project, the team member responsible for it created a main menu and developed player movement examples. They also focused more on exploring the LibGDX game engine, as these steps did not depend on user requirements. Meanwhile, the other two team members began developing them. Another team member started to work on the website while concurrently collaborating on the architecture with another team member.

Week 4

During the week from March 7th to March 13th, our team focused on reviewing all the tasks required for the project and assigning team members to work on deliverable documents. With an emphasis on producing documentation deliverables, two team members continued working on the User Requirements document, while another two started working on the Risk Assessment and Planning document. Additionally, we employed a Responsibility-Driven-Design approach to determine the main themes and components of our game. This involved creating the first prototype of our architecture using Class-Responsibility-Collaborator cards, utilising defined Candidates. This allowed us to lay the groundwork for proper implementation starting from the next sprint. However, during this sprint, we encountered a setback when our initial attempt to load a tile map failed, resulting in a backlog for the following week. Despite this, the issue was not critical and did not significantly slow down our progress. On another positive note, significant progress was made on our project website, with it nearing completion by the end of the sprint.

Week 5

During the final sprint, which started on March 14th and finished on March 21st, we focused on finalising all deliverables and preparing everything for submission. This sprint was arguably the most intensive one, as we aimed to complete all remaining tasks. The main focus of this week was on game implementation, which included making player movement work, fixing problems with the tile map, adding all interaction parts, and ensuring the collision system was functioning correctly. Additionally, we worked on implementing the end screen and score calculation. We also finalised our User Requirements and Risk Assessment documents and focused more on finishing the Architecture and Planning documents. Despite the intensity of this sprint, our team remained committed to delivering a high-quality product within the project's time constraints.

Overall, our project progressed very smoothly without any major issues with task completion. This was largely due to the effective team organisation and the responsibility of each team member towards their assigned work.

References

[1] I. Sommerville, “Agile planning” in *Software Engineering.* Harlow: Pearson, 2016, pp. 680-682.

[2] K. Schwaber and J. Sutherland (2020, Nov.). *Scrum Guide.* Scrum Guides [Online]. Available at: https://scrumguides.org/scrum-guide.html [Accessed: 10 March 2024].