

# Project Review Report

As we encountered new challenges throughout the course of the assessments our teams adapted to the new challenges. The team structure has not been modified much, because the original allocation of team roles[1] suited the individual skills of each team member and the positions aimed to value the strengths of each person. Additionally, every member seemed to be satisfied with their role and no request to change role was ever made. Besides adding a Version Control Manager in Assessment 2, when it became clear that maintaining the code we were producing was critically important [2], team roles did not change throughout the project. However, during Assessment 3 when the bulk of the programming was done, the Head Developer undertook a sort of a secondary Leader role, assigning tasks and organizing sprints as he had the skills and knowledge to breakdown the development process into manageable tasks.

Team management developed and changed the most during the different phases of the project. As part of our self-analysis process we assessed the team evolution with the CMM standards[3]. Now the project is finished it can be said that the team reached the *optimising level*, being able to cooperate and work efficiently towards the tasks set for each sprint. This was not the case when the team was formed, originally we considered ourselves at the *initial level*, due to the lack of organisation and experience with software engineering.

As we completed Assessment 1 we reached the *repeatable level*, since we established basic project management. But the team still struggled to work effectively and meet the deadlines. In Assessment 2, when everyone became more comfortable with the team environment and tools, we felt we reached the *defined level* and this was proven by an increased cooperation and overall quality of work. We reached the *managed level* in Assessment 3 when we swapped environment to work on another team's product and were still capable of working effectively, adapting our approach and methods we established during the previous phases.

An example of how the management evolved was during the final weeks of Assessments 2, 3 and 4 the team split into two smaller teams, where one focused on implementation and testing, and the other worked on documentation and the team website. Smaller teams proved to be easier to manage and allowed us to progress faster. This approach was not part of the original planning, but developed out of necessity to meet the submissions deadlines.

The one aspect the team struggled with the most and can still improve on is effective communication. It must be said that communication was generally good during meetings, when it was easier to check on the progress made and to clarify any misunderstandings or issues. However, meetings could not always be organized, especially during the breaks between terms. So the main mean of communication was Facebook Messenger.

Unfortunately messaging proved to be a very inefficient way of discussing the team's work, since not all team members could or would join the conversation at the same time. Additionally, it seemed that some members were just very hard to contact and get hold of via message. Several approaches were tried to achieve a better written communication and some advancements were made, but it's clear there still is room for improvement.

A possible solution would be to schedule conference calls so that discussions could be carried out in a similar fashion to a meeting, but without the need of being in the same place.

## Methods and Tools

The choice of using an Agile approach, in particular the Scrum framework, proved to be valid as it allowed us to create a product that meets all requirements in a relatively short amount of time. Given the nature of the SEPR project, where requirements were meant to change and the actual software had to be swapped between teams, we felt that the Agile method would be the most suitable one. Since it focuses on incremental development and promotes a flexible work environment, where the implementation should be easily adaptable to any changes. Therefore our choice remained consistent throughout the entire project.

Additionally, given our lack of real software engineering experience, the agile method allowed us to focus on producing small releases and therefore made it simpler for us to learn and improve our software engineering skills.

Scrum was chosen because it suited well small teams and also made team management easier, by being able to split work into several tasks and organize them in sprints. The framework has been used for the entire project because we felt that it allowed us to make progress fast, due to the ability to track and review our work through sprints.

It must be noted that at the beginning of the project our ability to break down tasks in an effective way was weak, but as we learned and familiarized both with the framework and with good software development practices we were able to improve our workflow and set up sprints that we could complete in time. The length of each sprint was also tailored to the kind of tasks required by each phase of the project, so development could be adapted as needed.

Our initial selection of tools[2] also remained consistent since each choice proved to be reliable and satisfied the team's needs without issues. Additionally once the team familiarized with the work environment, we felt that swapping a tool for a potentially better solution would just slow down our progress because it would force us to learn a new one. However as we moved through the different phases of the projects we had to select new tools to enable us to fulfill our needs. The following table summarises the main tools that were employed:

Tool	When it was used	Why it was needed
Jira	Entire project	Fundamental to track and manage sprints. Used also to store a backlog of tasks for the entire project.
Google Drive	Entire project	Needed a reliable suite of tools to allow collaboration on documents and easy file sharing.
GitHub	From Assessment 2 onwards	Needed a repository to safely store our code and track changes to it.
Facebook Messenger	Entire project	Needed a quick mean of communication to contact team members outside of meetings.
IntelliJ IDEA	From Assessment 2 onwards	Needed a reliable environment to develop the software and produce the code.
Photoshop	From Assessment 2 onwards	Needed a tool to produce the required graphics and assets for the game and website.
StarUML	Entire project	Needed a tool to produce clear UML diagrams to represent the software architecture.

# References

- [1] SEPR "Method Selection and Planning" Risky Developments [Online]. Available: <http://www.riskydevelopments.co.uk/documents/Plan1.pdf> [Accessed: Apr. 21 2018]
- [2] SEPR "Updated Method Selection and Planning" Risky Developments [Online]. Available: <http://www.riskydevelopments.co.uk/documents/UpdatedMethodSelectionAndPlanning.pdf> [Accessed: Apr. 21 2018]
- [3] Search Software Quality "Capability Maturity Model (CMM)" [Online]. Available: <https://searchsoftwarequality.techtarget.com/definition/Capability-Maturity-Model> [Accessed: Apr. 29 2018]