Project Review Report

SEPRet Studios

Anthony Nel, Ben Silverman, Jake Billington, Leah Moss and Shouyi Yuan

Team Management and Structure

In order to build an effective team that would complete the tasks we were assigned for each assessment, we knew that we would have to assign different roles to different members of the team [1]. By assigning people to roles, and communicating between these roles, we can work more efficiently (like a divide and conquer strategy) since members would know exactly what job they needed to do and there would be no confusion or conflicts [1].

At the start of the project, we had limited knowledge of each other's skills so we started by assigning basic roles, such as programmers and documentors, based on people's preferences. As we gained a deeper understanding of the tasks we had to complete we realised we would need more roles, such as designers, testers and a manager. As we began working together, these roles were altered slightly since realised team members worked better in certain roles. By putting particular members in certain roles, we were able to work more efficiently. This is because some people had a better aptitude towards coding or more experience coding than other members in the team which meant that part of the project/assessment would be completed faster. This allowed more time for other tasks such as documentation and allowed us to more frequently evaluate and improve the work we had done.

We used these roles in a hierarchical way [2] by applying each role to different subtasks. For this project, the documentors are in charge of specific pieces of documentation, the programmers are in charge of the software, the designers are in charge of the assets and the testers in charge of testing the software and documenting the results. The manager's role is to coordinate between these subtasks by having regular communication to make sure each subtask is completed correctly and managing when they should be completed by which ensures the team is working effectively [1]. When the manager had to decide when each subtask should be completed by, they had to consider the dependencies of some subtasks. For instance, the coding needs to be complete before the testing and the implementation report. The importance of the manager is one of the reasons we decided to add this role into our team structure [1].

Software Engineering Development Methods and Tools

For this project, we took careful consideration when choosing the type of software development methodology we wanted to use, since they tell us how to plan, execute and manage the development of our project [3].

We decided to use an agile method compared to traditional or hybrid methods for a few reasons. We are a single group consisting of 5 members and statistics show that projects consisting of less teams and less members are more likely to be developed using an agile method [3]. Large companies with many employees, high budgets and highly critical projects are more likely to choose a traditional method [3] because they have much more thorough planning and coordination [4]. Since we have a low budget and a small team we don't need to plan as thoroughly which in turn reduces the amount of time it takes us to develop the project. In addition, our project is not critical in terms of safety so reduces the amount of risk and preparation we have to do. This means agile methods were a better choice for us since we are only given a short amount of time to adapt our software for each assessment and we are constantly having to respond to the changing needs of the project [4].

At first we stated that we were going to use an agile methodology but once we learned more about software engineering we learned that there are many different agile methodologies. We decided that we should use Scrum. In Scrum there is an initial planning phase (like in most methodologies) [5] where the team will consider what requirements, architecture, methods etc that the project will require. Once this is complete the key feature of Scum can be used; sprints. Sprints are short, usually 1-4 weeks, phases of development [5]. There are frequent Scrum meetings which are led by the Scrum master (the person who adopted the manager role), and in each meeting a list of tasks, known as the backlog, is updated and tasks are delegated for team members to complete before the next Scrum meeting [5]. Once a sprint is complete, that iteration of the development is complete.

We chose Scrum over other agile methods because its iterative development supported how our project needs would change for each assessment. Furthermore, Scrum enabled us to keep on-track and meet project deadlines due to the regular meetings and generating the backlog of tasks. Other popular methods such as Extreme Programming (XP) did not support our team structure (stated above) since we separate the project into subtasks and it focuses too much on creating simple and fast releases [4]. Feature Driven Development (FDD) would mean we would have to focus on developing one feature at a time rather than working concurrently which again did not support our team structure (stated above) [4].

Before the sprints, we planned and stored our documents on Google Drive where we could continue to update and add to during the development phases. Scrum meetings were arranged using Facebook Messenger and during each meeting we would use a website called Trello to keep track of and update the backlog. For development we used the IDE IntelliJ with the LibGDX development framework and all code was stored on GitHub. With regards to design, we used Tiled to create the map and fortress assets and the website Piskel to create other pixel art for assets.

References

[1] WS. Humphrey, "The Team Software Process (TSP)," Encyclopedia of Software Engineering, January 2002. [Online]. Available: https://resources.sei.cmu.edu/asset_files/TechnicalReport/2000_005_001_13754.pdf . [Accessed Apr. 21, 2020].

[2] H. Van Vliet, Software Engineering: Principles and practice, 2nd ed. Chichester, England: John Wiley & Sons Ltd, 2000, pp. 97-98. Accessed on: Apr. 21, 2020. [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.128.2614&rep=rep1&type=pdf

[3] LR. Vijayasarathy, CW. Butler. (2015, Jan). “Choice of Software Development Methodologies: Do Organizational, Project, and Team Characteristics Matter?” IEEE Software [Online]. Vol. 33, issue 5. Available: https://ieeexplore-ieee-org.libproxy.york.ac.uk/document/7006383

[4] MA. Awad, “A Comparison between Agile and Traditional Software Development Methodologies,” Honours program thesis, University of Western Australia, 2005. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.464.6090&rep=rep1&type=pdf

[5] L. Rising, NS. Janoff. (2000, July). “The Scrum software development process for small teams” IEEE Software [Online]. Vol. 17, issue 4. Available: https://ieeexplore-ieee-org.libproxy.york.ac.uk/document/854065