Module	SEPR
Year	2019/20
Assessment	4
Team	Dalai Java
Members	Jack Kershaw, Max Lloyd, James Hau, Yuqing Gong, William Marr, Peter Clark.
Deliverable	Project Review

The team management we had adopted at the end of the project was very structured, with each team member having specific roles and a team leader overseeing all tasks to make sure they were completed. As we were using the Scrum framework [1], we nominated Jack as the Scrum Master, with the team as a whole becoming a self-organising Development Team. Whilst Scrum does not recognise sub-teams within the Development Team, we believed it was important for each member to be assigned a particular role, and that these roles would be flexible within the Development Team; Jack and Yuqing were responsible for implementing the main sections of code, Will was responsible for the artwork, Peter was responsible for the testing, Max was responsible for the website and James was responsible for the evaluation. Whilst these were the main points of responsibility for each team member, we made sure that the whole team were constantly aware of how each section was progressing via frequent group meetings (or scrums) where any ideas or modifications could be brought forward.

As per the Agile Manifesto [2] we were constantly reviewing and evolving our team structure and behaviour in order to maximise our effectiveness as a team. This allowed us to mature as per the Capability Maturity Model [3]; beginning in the Initial level in which our plans were unorganised and very dependent on individual team members and progressing to eventually reach the the Optimising level in which we were constantly seeking to improve our team management structure through measurements of software quality and group feedback on the effectiveness of previous structures.

Initially we agreed on a flat team structure, in which team members unanimously agreed on every decision made. However, we soon discovered that this structure was ineffective as, without a designated leader, it was difficult to assign tasks to team members and to hold team members accountable when tasks had not been completed by pre-defined deadlines. Therefore we made the decision to appoint a leader, which in this case was Jack. We then assigned specific roles to each member of the team, which changed based on the changing needs of the product and the artifacts which needed to be produced. For example, when a change report was required in Assessments 2 and 3, James and Max extended their roles from Assessment 1 to both make the required changes to their documents and explain the rationale behind this in the Change Report.

We also changed how we produced the artifacts; initially, two team members worked simultaneously on one section of the report, however as the project progressed, we realised that this produced inconsistencies within the same section. It also made it more difficult to ensure that all sections of the report were consistent. Therefore we made the decision to assign each team member a particular section of the report, have another team member proof-read this and then proof-read the entire completed report at the end of each iteration. This ensured each section of the report was consistent. Also, as we bonded as a team we were able to gain a better understanding of the strengths and weaknesses of each of our team members, meaning that we could assign more suitable roles to each member of the team. For example, we assigned Max the role of developing the website due to his previous experience with web design, and switched Will's role from software development to graphics as it became clear that his strengths lay in this field.

In terms of software engineering methods, we took an Agile approach to software development, in which working versions of the software were produced at regular intervals. More specifically, we used the Scrum framework [1]; every week we held a sprint planning meeting in which we determined what could be completed during the week and how this chosen work would be completed; this usually led to assigning team members roles based on their individual strengths, and every day we checked in with each other via a daily 'scrum' to determine what had been completed and what still needed to be completed. After each sprint, we reviewed our progress and future goals, and used this to help us plan the next sprint.

Whilst our plan from the start was to use the Scrum framework, this was initially not implemented effectively. As we had no designated leader, it was difficult to organise each sprint effectively and this meant that daily scrum meetings often did not occur. Upon assigning Jack as leader, we were able to effectively plan each sprint in detail and ensure that the sprints kept to their designated time frames. At the start of the process we also decided to incorporate RUP (Rational Unified Process) into our software development process; however, we quickly realised that this process was not suitable for our project as it focussed too heavily on documentation and made it difficult to incorporate changing requirements. As the project evolved, we were able to refine the structure of our sprints and implement improvements via Sprint Retrospective meetings. For example, as Assessment 3 only had a three-week time period, we reduced the time frame of each sprint and made daily scrum meetings a priority to ensure that our project was finished on time.

As we were using Scrum, we also had a Project Backlog which listed everything which was known to be needed in the product. As the requirements changed between assessments, this backlog also evolved to encompass these new requirements; for example, the need for aliens to move towards the Fire Station in Assessment 3 and the need for power-ups to be implemented in Assessment 4. Within assessments, we also refined items in the Product Backlog that were to be completed in the next sprint in order to give more detail and estimates of deadlines for items to be completed. As we matured as a team, our definition of "done" in these sprints was also refined with stricter conditions in order for us to produce a higher quality product.

In terms of software engineering tools, we used a GitHub repository in order to store different versions of the code and to allow us to collaborate on the same project. We used Discord to communicate and to hold meetings remotely, IntelliJ as our development IDE and Google Drive to store our documents. The reason for using Google Drive was to allow all team members to see whenever a document had been updated so that they could update any other relevant documents accordingly; this allowed us to follow the Agile principle of having dynamic documents. In our first assessment, we had less of a focus on storing documents on Google Drive; we found this was an issue as this meant we were only able to view some documents on the day before the deadline, making it very difficult to proof-read these documents and edit the other documents accordingly.

In our first two assessments, we also used a Trello board in order to organise the tasks which needed to be completed. However, as a team we found that this board was often unattended as team members did not update it once they had completed a project. Therefore, we adapted our approach to using a checklist created on Google Sheets as this served the same purpose and made considerably more sense for a small project such as ours.

Bibliography

[1] K. Schwaber and J. Sutherland, "Scrum Guide | Scrum Guides", Scrumguides.org, 2017. [Online]. Available: https://www.scrumguides.org/scrum-guide.html. [Accessed: 25- Mar- 2020].
[2] "Principles behind the Agile Manifesto", Agilemanifesto.org, 2001. [Online]. Available: https://agilemanifesto.org/principles.html. [Accessed: 14- Feb- 2020].
[3] M. Paulk, B. Curtis, M. Chrissis and C. Weber, Capability Maturity Model for Software, Version 1.1. Pittsburgh, Pa.: Research Access for Software Engineering Institute, 1993.