Project Review Report

Mozzarella Bytes

Assessment N°4

Kathryn Dale

Daniel Benison

Elizabeth Hodges

Callum Marsden

Emilien Bevierre

Ravinder Dosanjh

Project Review Report

While completing this SEPR project, we found that our team structure constantly evolved and changed to enable us to successfully counter the various problems that we faced (including team member absence, changing product requirements and the Coronavirus Pandemic). To enable us to successfully counter these issues as a team, the responsibility of project management and team leadership shifted to whomever was most able at the time. However, external factors - such as university holiday and the Coronavirus Pandemic - led to us having a much flatter team structure, as despite being unable to meet in person, team members had fewer other work commitments and so could have much more frequent video calls and communication to coordinate our work with the rest of the project.

We initially used a SCRUM approach, however by the end of Assessment 2 we were finding it hard to continue with regular SCRUM meetings as other work, module deadlines and a large amount of team member illness came in the way of our collaboration. Our approach naturally developed into a wider Agile approach, whereby we still prioritised client needs and each team member remained self-organising [1], but we didn't continue with SCRUM meetings every two days.

We found team member illness was the greatest risk to our project [2], as from early on we had team members unable to work due to illness, causing tension and stress between those members still present. This led us to distribute responsibility for each section of the project more widely, mitigating the impact that this risk was having.

By Assessment 3, the project was much more heavily weighted on the product and game itself over documentation, and so the management of the project became the responsibility of the heads of development, as they had a greater understanding of what needed to be done by when and by whom. We moved to a Belbin team model as this was the approach taken by the team before us. This approach was much more appropriate for us as each team member by this point had a much stronger understanding of the strengths and weaknesses of their co-workers. The flexibility of this approach allowed us to fully utilise each team member's strengths as the roles were less rigid. Furthermore, as we had already parted from a SCRUM approach, these new team roles helped group members to gain a better understanding of where their focus should lie. A Belbin team model was not something that we had come across before in our research of team management methods and we found that not only did it help us in this particular project, it also broadened our understanding of how different software development teams are able to work together and utilise each other's strengths while also recognising and allowing for each other's weaknesses [3].

For the duration of Assessment 4, the pace of development slowed as we did not have access to the faster University resources such as their more powerful hardware and greater variety of softwares as a result of the Covid-19 pandemic and subsequent government lockdown. We were fortunate in that the graphics had already been completed and we just needed to improve the code and documentation, however a slower network connection made the rate of sharing improvements and progressing with the project at the speed we had before very difficult. We overcame this by setting ourselves an earlier deadline than the official one to allow for delays in uploading our project and ensuring all documents were complete.

Despite moving away from a SCRUM approach to team management and roles, we kept the key features of a SCRUM methodology while developing our project.

In terms of the tools used during our project, there was little divergence from our original tools used in Assessments 1 and 2. We chose to use IntelliJ as our integrated development environment, due to a greater team familiarity with it, over Eclipse. We kept with this environment throughout the project as it was easy to use and we found that it only added to our team efficiency - thanks to not having to learn our way around a new environment. Other software tools that didn't change for the duration of the project was our use of LibGDX (because it already had all the features we needed - and more - to create a successful game [4]) and JUnit for our tests. We used JUnit because it is a popular, robust testing method that can make a multitude of different test types. Furthermore we already had a basic level of familiarity with it as a team. A final tool that didn't change during our project was our use of a software called Tiled to create and edit the map. Tiled's easy integration with LibGDX made it a clear choice to use for the graphics of the map (both for our original game, and the one we took over in Assessments 3 and 4) and we found no need to change or develop the software used.

There were some tools that we transitioned to early on in the project - such as using GitKraken over Git Bash and Git Kraken Glo over Github project boards. These tools were changed as a result of research and trial and error during the coding phase of Assessment 2, by finding which tools worked best for us. Once we found tools that we were comfortable using and that were suitable for the task at hand, we did not change them as they worked well for our needs and saved team member's having to learn new software and tools.

One new tool that we did pick up as a result of carrying on work on a previous team's project was Magicavoxel. This tool took time to learn, however it allowed us to maintain continuity with our graphics and user interface with the game.

Bibliography

[1] Svitlana Rumyantseva, ‘How to Manage Software Development Team? Methodologies, Practices & Tools’ , *qubit-labs.* March 22, 2019 [Online] https://qubit-labs.com/manage-offshore-software-development-team/ [Accessed: 11 March. 2020]

[2] Team Mozzarella Bytes, Risk Assessment Document, https://db1218.github.io/KroyAssessment4/assessment4/overflow/Risks%204.pdf

[3] Belbin, ‘The Nine Belbin Team Roles’, *Belbin,* [Online] https://www.belbin.com/about/belbin-team-roles/ [Accessed: 06 April. 2020]

[4] libGDX, ‘Goals and Features’, *badlogicgames.* 2013 [Online] https://libgdx.badlogicgames.com/features.html [Accessed: 06 April. 2020]