

Method Selection and Planning

Team 27 - BlackCat Studios

Jack Vickers, Hubert Solecki, Jack Hinton, Sam Toner, Felix Seanor, Azzam Amirul Bahri

Software engineering methods

For this project, we have decided to use the Scrum method of development. We've chosen this method because we believe it would be the most efficient for us and enable us to get a lot of work done in a short period of time. We divided our team of 6 into a development and documentation team, each consisting of 3 members. We have had regular team meetings as a whole team and have also had separate meetings for each internal team. In these, we discussed what has been done since the last meeting (the beginning of a sprint) and what needed to be done before and during the next meeting.

Software development approach

As a team, we decided to follow a risk-driven approach based on the spiral software development approach. Rather than having system prototypes, each iteration (scrum sprint) for us consisted of completed sections of deliverables and implemented features of the game.

During the first iteration, we gathered requirements from our customer and then outlined the plan to split the team, so that documentation and development could be done simultaneously. We wanted development to be started early on so that we had more time to work on the game to attempt to reduce the risk of not completing the required features. We chose this approach rather than a more structured approach so that we could work on certain tasks as we needed to and could go back to previous tasks/ iterations if we realised that we missed anything or if any issues or major risks were identified.

Development and collaboration tools

- IntelliJ IDEA is the IDE that we chose for the development of our project. We've decided to use this because it has Git features and Intellisense and it is lightweight.
- We are using libGDX for the game engine because it is free, open source and works on a variety of platforms, which is important since our customer wants our game to work on Windows, Linux, and Mac systems. Plus, we are also considering the fact that our project could be continued by another group for assessment 2 so also chose this because the use of this engine is not too complicated and would also have been used by other teams already. This engine also has a lot of documentation which was useful for understanding how to use it.
- We used GitHub for version control so that features can be implemented by different team members and then merged. We each have our own individual development branches to work from.
- There is a SiteDevelopment branch which is used to deploy a GitHub pages website which is the "public face" of the project. This contains weekly progress updates and links to all the deliverables needed for assessment 1. We chose to use GitHub pages for the website so that it would be easy for another team to take over for assessment 2. To do this, they could clone the entire github repository and build the website from the SiteDevelopment branch.
- The website has been made using material for MkDocs, which assists with making a professional-looking static site from markdown documents. It allows the website to be deployed easily to github.

- Google docs and sheets have been used for documentation since they offer version control functionality which allows us to return to previous versions of documents if anything goes wrong. These also allow us to work on documents simultaneously on different devices. This was important for quick completion of deliverables.
- Plantuml.com has been used to create sequence, state, class, package and use case diagrams.
- Echeung.me has been used to make CRC Cards for every class.
- Discord has been used for communication and meetings during the Christmas break.
- We are developing the game on windows laptops

Alternative approach and tools considered

- We considered using the waterfall approach for our project. This required each member to come together and start doing each part of the project one at a time. This may have been more time consuming and possibly have required more meetings. We needed to use the most efficient approach in our opinion and that was why we chose the risk-driven approach.
- As for the tools, we considered using the Eclipse IDE as it is the most widely used tool to build a project. To make our project more unique and easily adaptable, we chose IntelliJ IDEA.
- JMonkey engine was another engine that we considered using. It has a community but it is not very well established and their documentation is not very structured and not as professional as libGDX.

The team's approach to organisation

We decided to separate into groups, and delegate the responsibilities of each section. As a group, we decided to nominate Hubert as group leader, as he had a clear understanding of the project and a good plan to complete the project. We also designated Felix as lead developer as it was clear he understood java to an advanced level and would be able to use this knowledge to vet the code and aid with finding bugs. Felix was assisted by Sam Toner and Jack Hinton as backup developers.

As the website was a lot of work, we decided we needed a website leader, and for this we designated Jack Vickers, who consistently updated the website with weekly team updates throughout the project.

Azzam Bahri was nominated as the lead for requirements, as he showed a clear understanding and enthusiasm towards it. He was assisted by Hubert Solecki and Jack Vickers as requirements backup and everyone else collaborated on it. This was a greatly beneficial organisation, as it allowed for the group to remain caught up, and make sure everything was going smoothly, while simultaneously having a sense of leadership within the section, guiding the documentation in the correct way. Using similar roles, Hubert, Azzam and Jack Vickers led and monitored the methods section and planning, as we believed it was closely linked to requirements, and everyone else in the group helped work on it, while being guided. This meant we would be able to achieve both quite quickly.

Sam Toner, Felix Seanor and Jack Hinton, worked together on the architecture documentation, with help from the rest of the group, working together and checking everyone was achieving what was required, and working together to understand the UML diagrams, producing them quickly and efficiently. Finally, Hubert Solecki, Azzam Bahri and Jack Vickers worked on risks, with Jack Vickers leading the risk analysis, and Hubert and Azzam assisting. They were helped by the rest of the group, developing a list that was studied closely by everyone involved throughout the project, to avoid any risks occurring.

We ensured there were always backups, and there was always someone to take over any part of the project in the event of anything going wrong, leading to a bus factor of 4 for most of the project. The bus factor of the website is lower than the rest, as it was a minor part of the project, with a bus factor of 2.

In the final 2 weeks of the project, the whole team worked on programming so that the game would be completed on time. The documentation team were given specific features to implement: Jack Vickers implemented the pause menu, Hubert implemented the start menu and Azzam implemented the Highscores screen.

Role	Name
Team Lead	Hubert Solecki
Team Lead Backup	Felix Seanor
DevOps Lead	Felix Seanor
DevOps Backup	Sam Toner, Jack Hinton
Requirements Lead	Azzam Bahri
Requirements Backup	Hubert Solecki, Jack Vickers
Risks Lead	Jack Vickers
Risks Backup	Hubert Solecki, Azzam Bahri
Website Lead	Jack Vickers
Methods and Planning Lead	Azzam Bahri
Method and Planning Backup	Jack Vickers, Hubert Solecki
Architecture Lead	Jack Hinton
Architecture Backup	Sam Toner, Felix Seanor

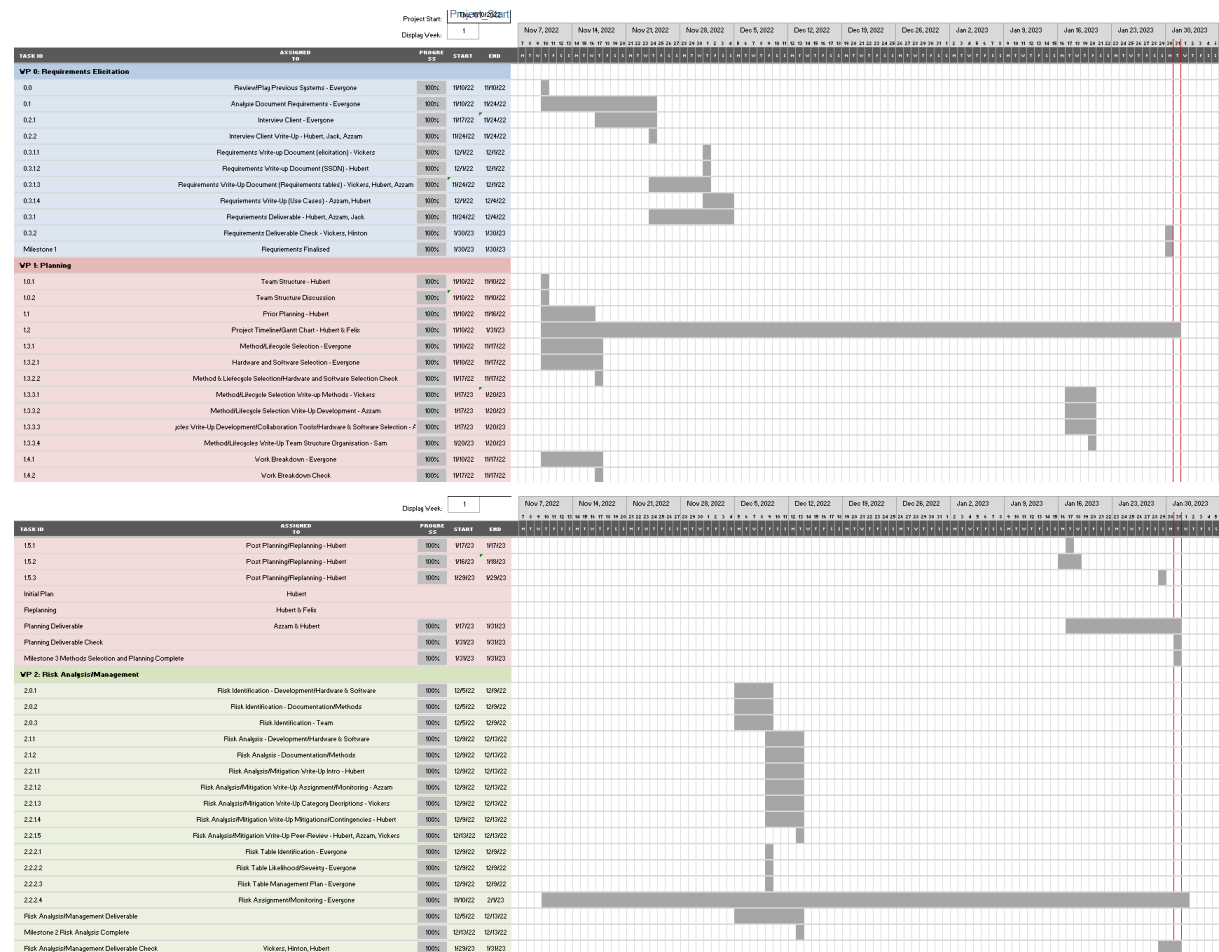
How the plan evolved overtime

At the start of the project, the team was meeting one/two times a week and worked on tasks that had been discussed and recorded on the project Gantt Chart as well as work packages and milestones. Over the Christmas break, we had fewer meetings as an entire team and decided to have smaller meetings in our internal teams instead. These were held over Discord. After exams at the start of term, we had meetings almost every day until the submission of the project. We decided to do this because we knew that there was still a lot of work to get done and that a lot of communication was needed to get things done.

After an initial plan at the start of the project that lasted till December, we had to reevaluate and replan for the December holidays to ensure deliverables were getting finished.

Additionally, we encountered times where we had to extend deadlines due to certain risks we encountered that are registered in the weekly updates on our website. Lastly, we had two further replans when we returned after exams to ensure we had a plan to finish the project and the game in time for the project deadline, and in the last few days covering the final parts of the project to finish.

Gantt Chart





Above is the Gantt Chart for the project. Please zoom in to view it better. An alternative to viewing the Gantt Chart is available on our website at the following [link](#). We have further described changes to the Gantt Chart along with the risks we faced that caused these changes on our website. One example is the risk R10 (missing individuals from meetings), which minorly pushed back deadlines by a few days occasionally. These were mitigated by using online communication services such as Discord (live chat and screen share) to enable good communication and collaboration. Further work was completed when we were together in person again where we acknowledged changes and finalised them.