Method Selection And Planning

ENG1 Team 7
Broken Designers

Adam Brown
Morgan Francis
Oliver Johnstone
Shabari Jagadeeswaran
Laura Mata Le Bot
Rebecca Stone

The method selection and planning report here seeks to offer clarity on the methods and planning stages for this project. In general, the approach we took over was quite similar to the previous assessment. We began this aspect of the project by gathering as a group and reviewing how we would look into the changes to be made in the report.

Our choice of development and collaboration tools was a heavily debated topic. In the end, we chose the use of discord for communication. This was decided on due to the fact that many team members had already used the tool, which helped reduce the time needed to set up our communications. Discord had limitations in terms of file sharing though, so we made use of google drive and docs to effectively organise and share our files.

Another area we deliberated on was our choice of IDE. The factors that went into this choice, included how well the IDE integrated Git's version control systems as well as how easy the project was to work on from the IDE. Our preferred IDE was intellij IDEA which had better integration with github than other options we considered, such as Visual Studio Code.

During the first meeting, we went through each of the other groups' presentations and discussed which would be the most suitable pick for the second part of the assessment. We tabulated our ideas, considering any issues we found in their code and eliminating any groups with major problems. This allowed each team member to add their own ideas and opinions and helped us come to a decision without conflict.

Documentation was another major consideration in choosing a project. One of our mistakes in choosing which project to take over was not giving this aspect enough weighting in the decision. The project's documentation was missing parts that would have helped us in our continuation of the project especially in areas such as the comments in the code. We also discussed some important factors of taking over a project and decided which group's project was better. We also had a look at the feedback provided to us from assessment one and took some points from it to make our change report.

After we made a decision on which group to pick, before beginning work on the project, we used the Sailboat retrospective method to discuss what went well during the previous assessment and how we could improve moving forwards. From previous research, the sailboat referencing is a fun and creative way for teams to reflect on the sprint and identify areas for improvement.**[1].**

Some key points we took away were to have clearly defined roles, shadows and to set firm deadlines in the future. Our shadows were team members that had worked on certain deliverables from assessment one, mentoring and helping the new member responsible for this in assessment two.

During the first half of the project, we used Gantt charts which were useful but time consuming to make. We decided this time was better spent in other areas, so in order to combat this issue in assessment two, we used Trello. This is a project planning website that allows groups to make boards so that they can work in a systematic manner. We set up particular tasks for each and everyone in our group alongside a deadline for when the task must be completed by.

Since pair programming was very successful for us in assessment one, we decided to continue with this approach. This is a software engineering method that involves two developers working on one workstation. It helped us get past minor hiccups where one person's knowledge around a particular aspect of implementation was stronger than the other.

Using a version control system is important to allow us to access previous versions of the game when required. We considered both git and SVN, settling on git as our preferred system. As this is a distributed version control system, we could all work on local copies independently. It doesn't allow for strict hierarchies as SVN does, however this wasn't necessary for our project.

To create the UML diagrams for our architecture report, we chose to use plantUML. This allowed us to write code that generates diagrams. This was a better alternative to drawing out all the diagrams by hand as it was easier to use a text-based approach. We could easily add new sections to the code and generate the UML again, rather than needing to move around sections in a diagram we made ourselves. An alternative to this was the tool mermaid. This is JavaScript based and inspired by Markdown, which we weren't familiar with, so we decided to use plantUML to generate our diagrams.

We followed a routine where we met every Monday and Wednesday to discuss our plans for the week and complete some parts of the project before we met again for the practical session the following week. Doing this set us into a consistent rhythm that resulted in an even workflow of the project over the term. This planning technique helped evaluate stress caused by unforeseen problems as we always had spare time available.

The team adopted new planning techniques such as using Trello and pair programming after reflecting on the initial assessment and identifying opportunities for improvement using the Sailboat retrospective method. Overall, the process of choosing a strategy and planning was an integral part in completing the project together as a group.

Bibliography

[1] Tech Agilist, Sailboat or Speedboat–Sprint Retrospective, Tech Agilist, [Online].

Available:Sailboat or Speedboat - Sprint Retrospective - Tech Agilist [Accessed: 19 April 2023].