Engineering 1 Group Assessment 2
Architecture
Cohort 3 Group 23

Team Members:
Harry Draper
Seif Hussein
Tikhon Likhachev
Thomas Maalderink
Joshua McKean
Sebastian Armstrong

Methods and Tools

For our project, we will be using agile development methods. These will allow flexibility and easy changes to our development process. The ability to keep changing requirements enables us to review our project and make changes to the requirements and timescales when needed. This helps us to fit our requirements well and adapt to circumstances that could affect project delivery or planning.

The main tools we have used in our project are:

- Discord for communicating with the group outside of the meetings.
- GitHub for task management, website development and code development.
- Google for writing up documents and managing files.
- PlantUML in IntelliJ and Lucid for creating diagrams.

Our team used Discord to communicate with each other and plan anything outside of meetings. This is because everyone was familiar with it and already had accounts, making it the easiest option. Discord allows us to create channels focused on different topics, meaning that people were quickly able to find relevant information..

Our group used GitHub for source control and code development/maintenance as well as feature management and reviewing any tasks. GitHub is very useful because it lets everyone work on code simultaneously. Team members could create their own branches and then push/pull changes to and from the main branch. This allowed the coding team to work on the code individually and then combine it all without the risk of data loss. We turned on mandatory code review and disabled committing to main. This meant changes were checked before being committed, meaning that there were no overlaps in the code.

We used GitHub Pages for the website. This is because we already used GitHub for code storage. The group member creating the website was familiar with GitHub pages, so we agreed it would be the easiest way to create the website. The page was hosted directly from the GitHub repository. This kept all source code in a single repository and reduced the “bus factor” while making hosting easier.

We created a Google drive to store all other parts of the project. Everyone on our team was familiar with Drive and it allowed everyone to access every document. Google Drive also allows everyone to see changes that are made to files and provides version history. This mitigates the risk of data loss and prolonged sickness.

Our group chose to use Google Docs to write documentation for the project. Docs allows everyone to have easy access to all the files to edit while keeping version history.
The team used Lucid and plantUML to create diagrams. Lucid is an online tool for the collaborative drag-and-drop generation of flowcharts and diagrams. We used it to generate our class UML diagrams. This worked better than an alternative we

considered, Mermaid, as it was online and graphical. Mermaid would have required installing software to generate images. To generate our Gantt chart, we used plantUML's online generator tool. plantUML allowed us to assign people to specific tasks on a Gantt chart, helping us visualise who was free to take more work. We backed up our plantUML code in a Git branch and tried to set up a GitHub Action to generate the image from the plantUML source on commit, but we were unable to get it working and decided it was not worth the time.

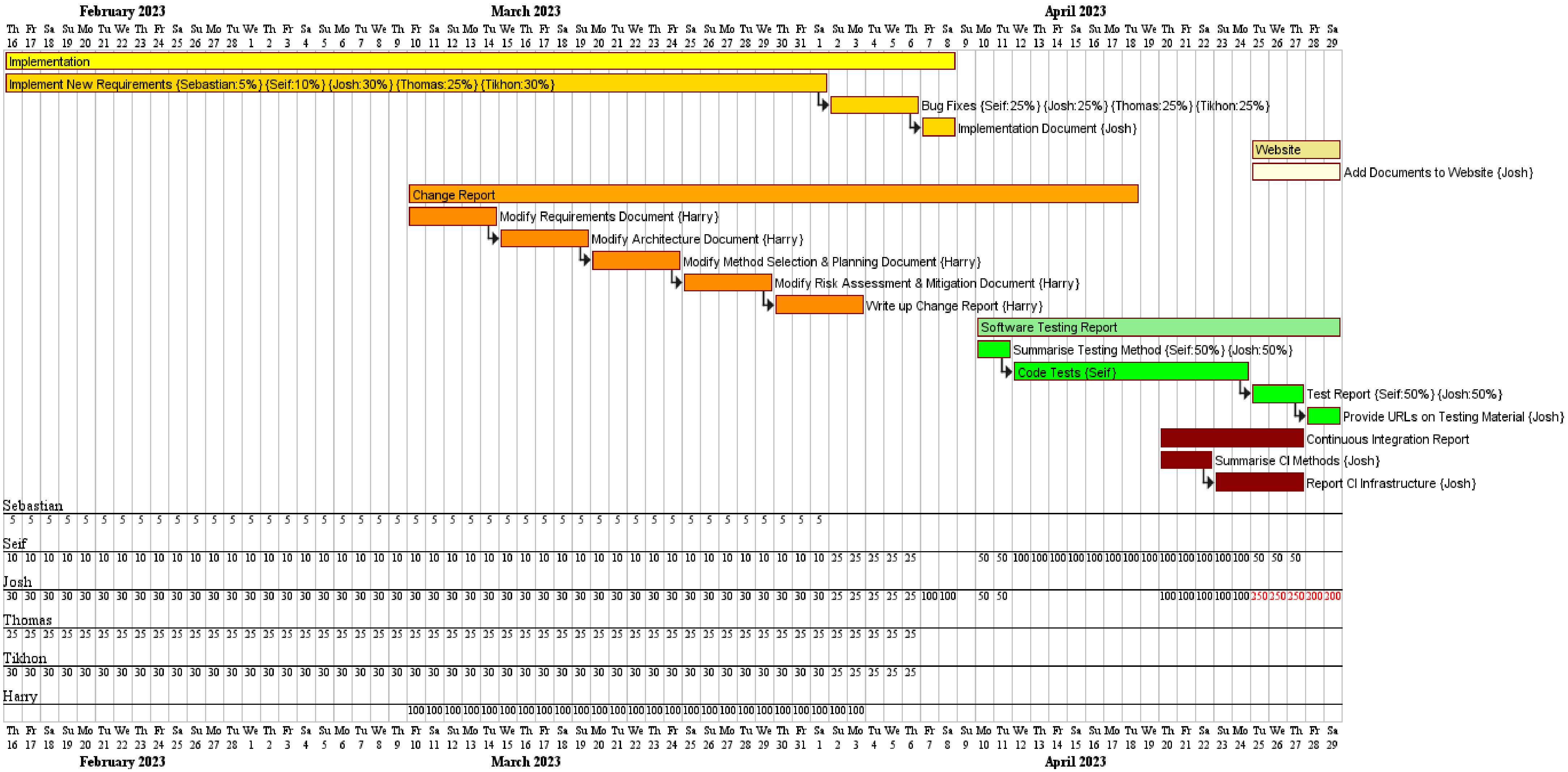
Team Organisation Approach

Our team organised meetings through our Discord server and in person.. In meetings, we reviewed our progress on the tasks and work packages. We assigned people tasks during the meetings or for their own time. This was appropriate for the team as we are busy students, allowing us a healthy mix of collaboration time and self-led work we could do when we were most productive.

We assigned each person to different tasks throughout. Josh, Seif, Tikhon and Sebastian were set with doing the implementation of the new requirements and testing. Harry was responsible for the documentation including the change report and modification of the original documents to coincide with the new team's methods and additions to the code.

Frequent communication on Discord ensured that everyone was up to date with the project and all data was backed up on each of the teams devices to prevent the loss of any progress

Project Planning



After taking over the project the team reviewed the existing base plan created by the previous team. We used this to aid with completion of the project and set realistic timelines for it.

Using the agile development methodology, we edited the plan when we needed to in order to make it effective for our group. We had some issues with people being ill or missing meetings in person. We overcame these issues by keeping in contact with Discord. If people could not make it to meetings but were free, we would have hybrid meetings, which worked effectively. As we used Google Docs, everyone could access the documents. This meant that people could help out even when they were not at the meeting.

Key tasks:
These key tasks originate from the Gantt chart. The steps that we followed in our plan were: Implementation, Change Report, Software Testing Report, Continuous Integration Report and Website

Implementation:

- Document and provide code that meets the requirements for assessment 2.
- Provide an executable JAR file of the game

Change Report:

- Provide summaries of the processes and tools used to plan, track and review changes of the assessment 1 deliverables.
- Modify the previous teams assessment 1 deliverables to the assessment 2 deliverables.
- For each item of Requirements, Architecture, Method Selection and Planning and Risk Assessment and Mitigation, explain and justify the changes made and if there are no changes necessary explain why.

Software Testing Report:

- Summarise Testing methods and approaches and explain why they are appropriate
- Give a brief report on the tests including the statistics and the results achieved. If any tests fail, explain why they either cannot pass or why they failed.
- Provide precise URLs of the testing material on the website, comprising the results and coverage report.

Continuous Integration Report

- Summarise continuous integration methods and approaches, explaining why they are suitable for the project.
- Give a brief report on the continuous integration infrastructure set up for the project.

Website

- Update team website to include all assessment 2 deliverables as well as the previous teams assessment 1 versions

- Include an executable JAR for the game
- Include the version control repository of the team's code.