Method selection and planning

1) Give an outline and justification of the team's software engineering methods, and identify any development or collaboration tools that the team has used to support the project or the team working. Justify the fitness of the selected tools with the team's software engineering methods and discuss alternatives considered. (3 marks, ≤ 2 pages).

2) Outline the team's approach to team organisation, and explain why the chosen approach is appropriate for both the team and the project (2 marks, ≤ 1 page).

3) Give a systematic plan for the project. Your plan should lay out the key tasks, their starting and finishing dates, as well as task priorities. The plan should also identify a critical path and task dependencies. Provide weekly snapshots of the plan on your team's website and discuss how the plan evolved throughout the duration of the project (5 marks, ≤ 2 pages).

[Total 10 marks]

(1) An outline and justification of the team's software engineering methods

Our principal software engineering development methodology is Agile. We chose this because it copes well with volatile requirements, which we expect this project to have.

It employs short iteration cycles, with low documentation overheads, and focuses on maintaining an inclusive relationship with the client. This allows us to incrementally deliver working prototypes and get immediate feedback for the next iteration.

Agile dictates that face-to-face communication is better than online, but this is unfortunately not possible with the current pandemic.

(1) Developmental / Collaborative tools we have used, and (2) our approach to Organisation

Our codebase is stored in a repository on GitHub, which we felt was the clear leader for version control and enabling team-working on a shared programming project. Additionally, it has inbuilt integration with VS Code -- a working environment many of us use regularly -- and the platform in general has by far the most tutorials and useful resources available.

- Alternatives include Bitbucket by Atlassian or other repository hosting tools / version control tools, but we felt GitHub does both of these things best.

Primarily we collaborate on Discord, using the platform to communicate (both in calls and through text), share files, and self-structure both the assignment and current progress of deliverables.

We believe Discord is fit for this purpose for the following reason:

- We all had existing Discord accounts we could use, and were familiar with the platform and how to use it.
- Discord permits creating a custom server and modifying its structure to best suit the application, with numerous pre-built templates to speed up this process.
    - This allows the server to be flexible and accommodate multiple facets of the engineering process. The centralised approach it facilitates increases working efficiency and coordination, and reduces the complexity of working across multiple applications.
    - For example, we have text and voice channels for each deliverable, allowing uninterrupted communication between assigned team-members.

Possible alternatives we considered included Slack, Facebook Messenger, and Zoom.

- Slack is the closest proxy to Discord, however we felt that our lack of experience with it (and few of us having pre-existing accounts) was enough reason to choose Discord as they are very similar anyway.
- Facebook Messenger has both voice and video call options, but poor support for file sharing (including previewing embedded images etc.) and no ability to structure a groupchat in any way. Zoom had similar drawbacks.

A planning tool we use is Trello. This introduces the concepts of Kanban-style “Boards”, which can be created for projects and used to keep track of tasks.

- It allows us to keep track of task priority, status, and assigned members, along with a short description and a task deadline.
- The board is split up into “To-do”, “Doing”, and “Done”, which allows us to asynchronously self-organise and determine which [sub-]tasks need to be completed.

We chose Trello since a few of our members had experience with it already, and it is generally well-regarded as a project planning tool due to its intuitive UX.

A Systematic Plan for the Project

See Lecture 4 -- Project Planning and Risk Management