# ENG1 Team Project

## Deliverable 4: Method Selection and Planning

### Team 13 – Team Unlucky

LILLY BROWN
ROSCOE GILLATT
ADAM JOHNSON
MATTHEW TAYLOR
BRANDON WEST
AYMAN ZAHIR

a)

We chose to base our engineering methods off the Rapid Application Development model (RAD). Because of the way we chose to decompose the overall project for individual team members to work on, we didn't think it was realistic to adhere to the Waterfall approach. This would have meant one member would have to have strictly finished their section of the project before another member could move onto theirs, which would massively hinder development.

The Agile model would be more suited to the team's needs of flexibility, however it is not well suited to small projects, because for the implementation itself, there were not enough parts that the problem could be broken down into. Furthermore, the Agile method implies that the customer may be changing the requirements during the development process, which is not the case with this project.

The RAD model focuses on rapidly developing prototypes, until eventually reaching the desired requirements that were concretely laid down at the beginning and finally providing a concrete implementation. Sommerville states various tools and techniques that come with RAD [1], however we chose not to use these as they were not appropriate to our project, because they relate more to programming with a database. Beynon-Davies states *"The developers, after some initial investigation, construct a working model that they demonstrate to a representative user group. The developers and the users then discuss the prototype, agreeing on enhancements and amendments"* [2]. This is representative of what occurred in the weekly practicals, where we would have a prototype that was representative of the final product and then it would be critiqued by the client and we would go away and work on it for the next week.

The key software that we decided to use in helping us organise and communicate was Discord, this was because most people on the team were already familiar with Discord, how it worked, and already had it installed. We could have used Slack instead but we saw no real benefit to everyone installing and familiarising themselves with new software.

To organise meetings we used When2Meet as it is a simple website that serves its basic purpose. No account creation was required, team members could quickly type in their name, select their availability and see when other members were available - this made planning group meetings very efficient. We could have used traditional methods, such as simply asking in our Discord server when people would be free, but this rarely works because it is difficult to get the bigger picture from the availability people give.

We used Google Docs to manage and organise deliverables, because it has simple version control and edit history that can easily be rolled back. Furthermore, it is much better for synchronous collaboration in comparison to something like GitHub.

We did use GitHub as our main website and for software implementation control, as it keeps everything in one place that is clean, organised, and easy to navigate, with rollbacks and commit history too. Although there are alternatives to GitHub such as AWS CodeCommit, it would be illogical to choose this, because GitHub is so widely used within the programming and software engineering community. We decided it would be beneficial for us to learn how to use GitHub within a group development context and it was also much easier to learn, because of the available content online.

For Gantt Charts we used PlantUML as it was quick and easy to learn, and resulted in a fairly aesthetically pleasing chart that was easy to read. We could have opted to use Dia or diagrams.net, but we found that PlantUML was easier to use and there were more resources available for us to learn from quickly.

For game implementation we decided to use IntelliJ and LibGDX,although we could have used Eclipse instead, some of our members were already familiar with these libraries and architecture. This meant, choosing IntelliJ and LibGDX would reduce the amount of learning and research we would have to do overall, and having some people more experienced in the language would be a benefit to everyone in the team.

**References:**

[1] Ian Sommerville, Software Engineering, 7th ed. Addison Wesley, 2004. Chapter 17
[2] P Beynon-Davies, C Carne, H Mackay & D Tudhope, Rapid application development (RAD): an empirical review, European Journal of Information Systems, 1999, 8:3, 211-223, DOI: 10.1057/palgrave.ejis.3000325

b)

Sommerville states that within a team there should be a variety of three different roles [1], task-oriented, self-oriented and interaction-oriented members. Within the group there is a healthy balance of contributors that have different focuses; our team has two self-oriented members, two interaction-oriented members and one task-oriented member.

In addition to Sommervilles three personality types, we also assigned ourselves to one of Belbin's nine team roles, this would allow us to better understand each other's strengths, weaknesses and place in the team. None of the team members chose the same role and there was a balanced across the types of role chosen.Teamworker, Plant, Research Investigator, Shaper and Coordinator were roles the team selected; from this the team will have strengths that reduce or mitigate other weaknesses, for example, the Plant's weakness in communication will be reduced by the Coordinator's stronger interpersonal skills.

There may be both technical and administrative leaders within a group [1]; after being asked whether they would be interested in a leading role, two team members said that they would, with Roscoe showing an interest in technical leadership and Brandon showing an interest in administrative leadership. A conversation took place during a team meeting about the management structure and although we decided to keep it relaxed it was mentioned that both Brandon and Roscoe have naturally leaned towards leading roles. Sommerville also mentions that democratic leadership is more effective than autocracy, so decisions will be made as a team and not by an individual, with leading roles only serving as task delegation and general project management.

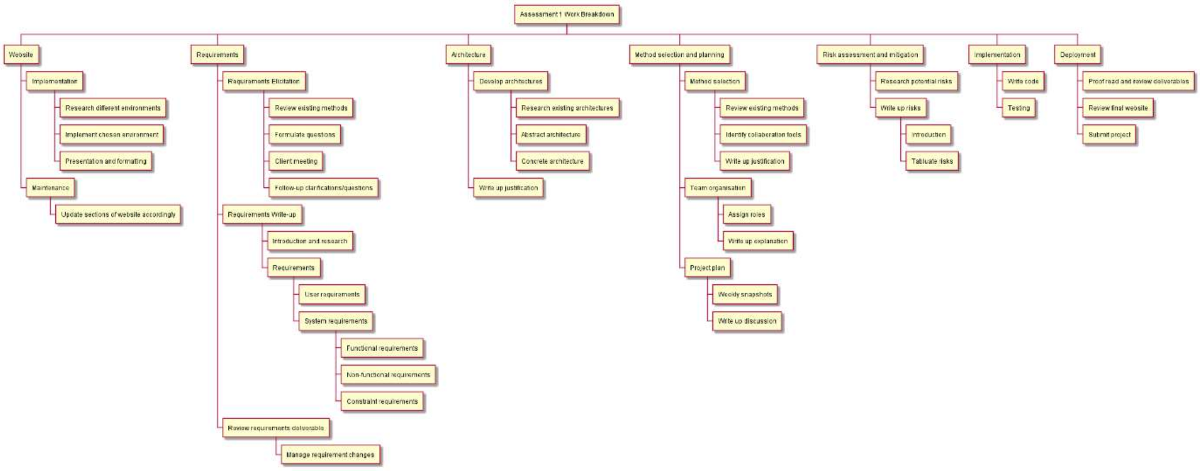
The main method of communication internally within the team is the Discord server that has been set up, this is split into five channels: a documents channel for sharing links and snapshots of documentation for the project, a research channel for sharing useful links for the team to research to aid the project, a module resources channel that contain the links to the resources provided by the module, such as the assessment brief, a channel for the delegation of tasks and finally a channel for general communication. There is also a voice channel for meetings that have to take place online.

**References:**

[1] Ian Sommerville, Software Engineering, 7th ed. Addison Wesley, 2004. Chapter 25

c)

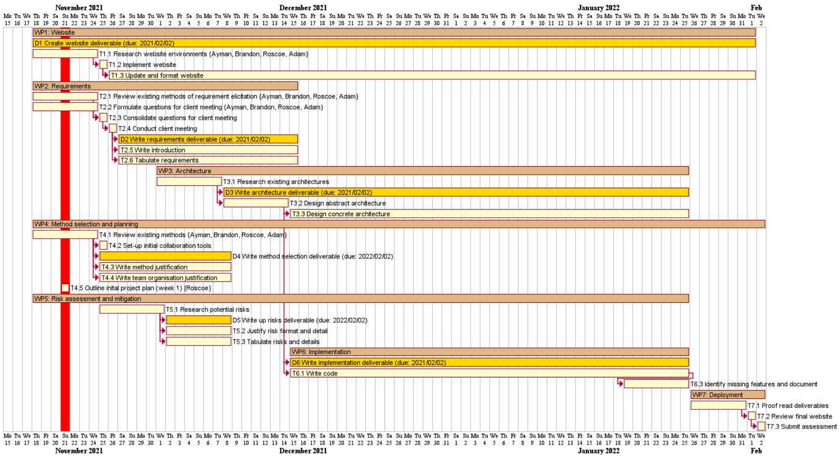
## Plan overview



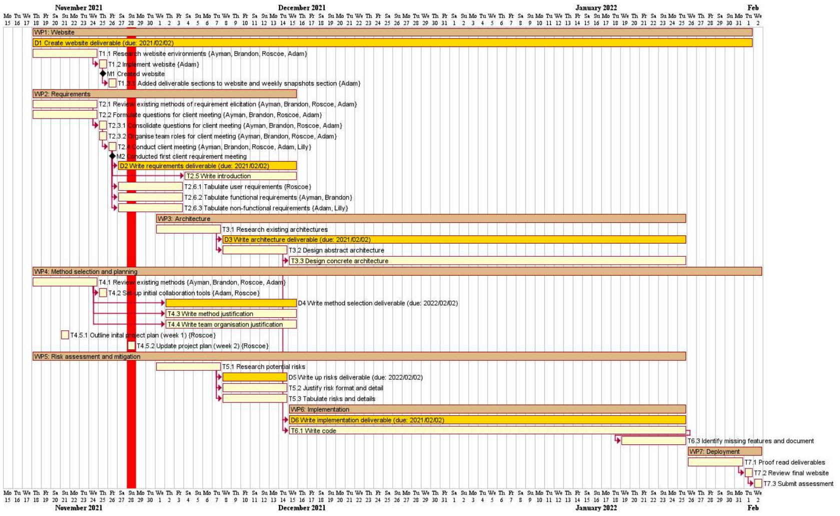
This diagram covers what tasks needed to be done for successful completion of the project.

## Weekly Snapshots

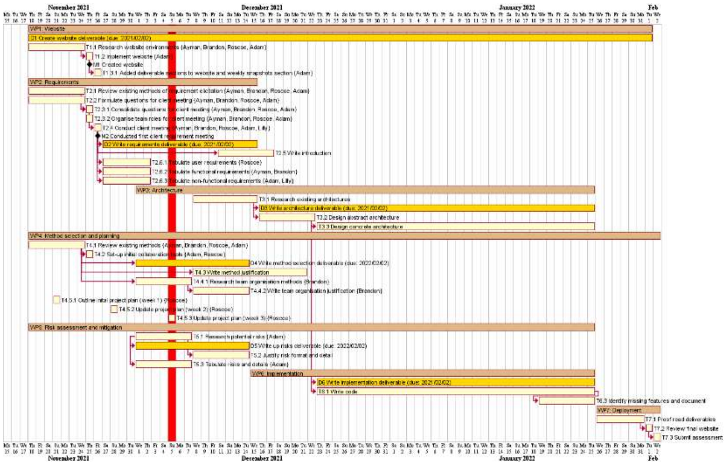
Each week a snapshot was made of the plan, which showed who was doing what task and what still needed to happen. These changed over time as the plan changed, due to various events occurring. All snapshots can also be found on the team website.



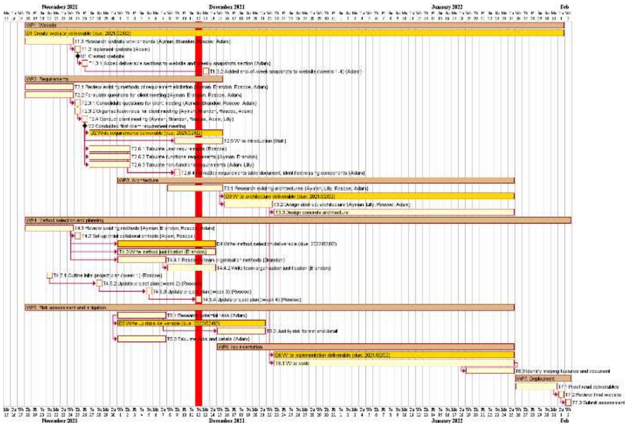
This is the initial plan for the first week, made as we were soliciting requirements

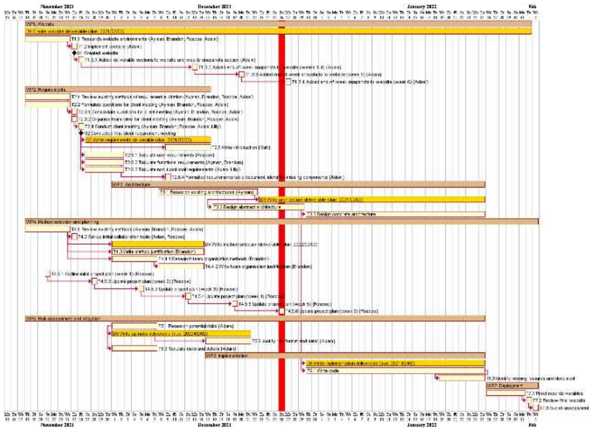


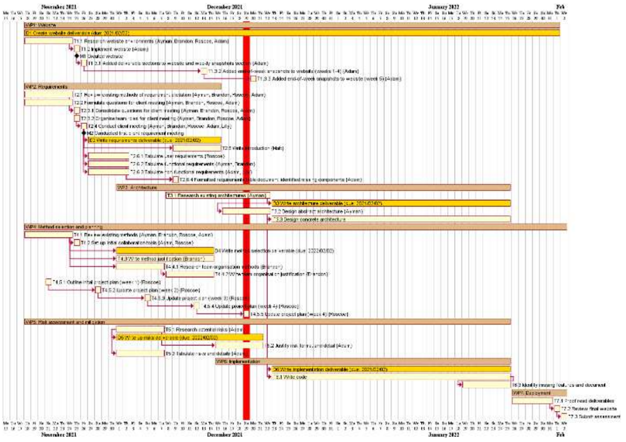
Once the team had an idea of what was happening, the plan could evolve into one with more detail

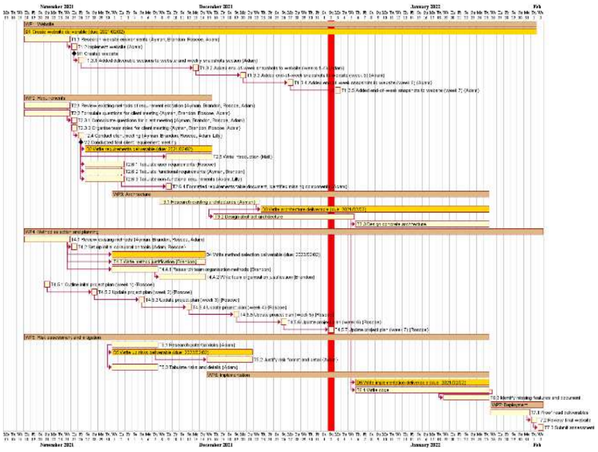


As skills of team members were explored, tasks best suited to them could be delegated for the best results.

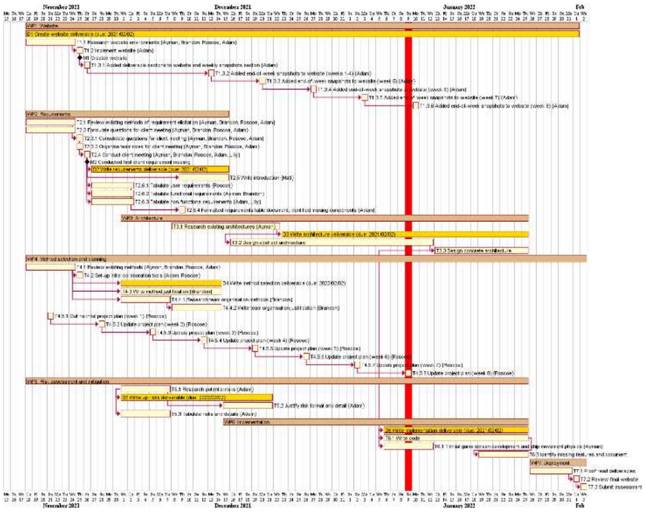


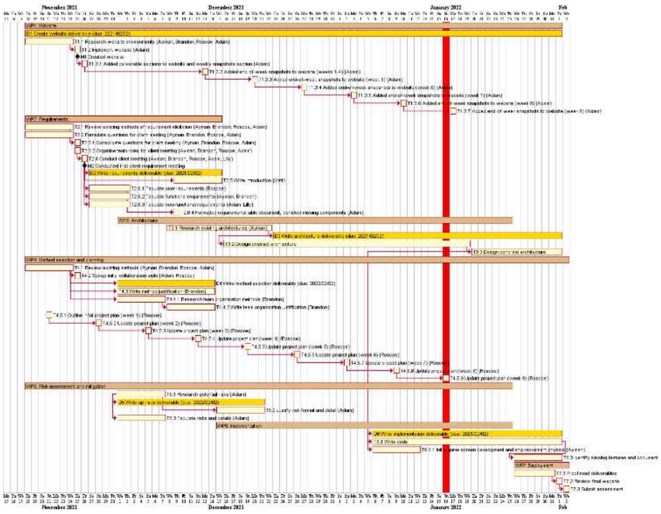


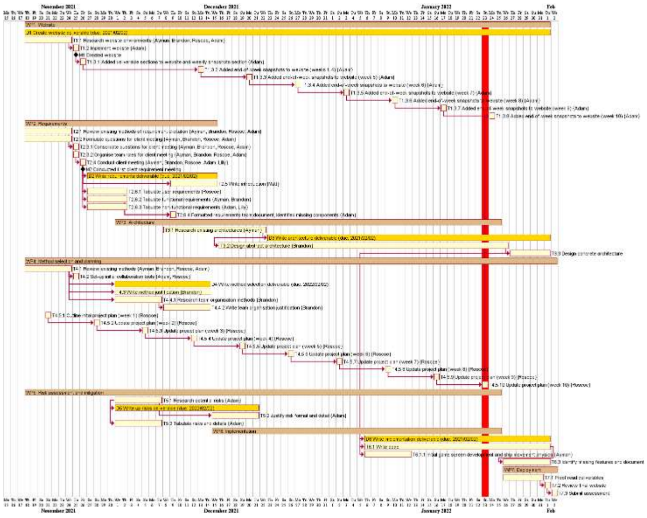


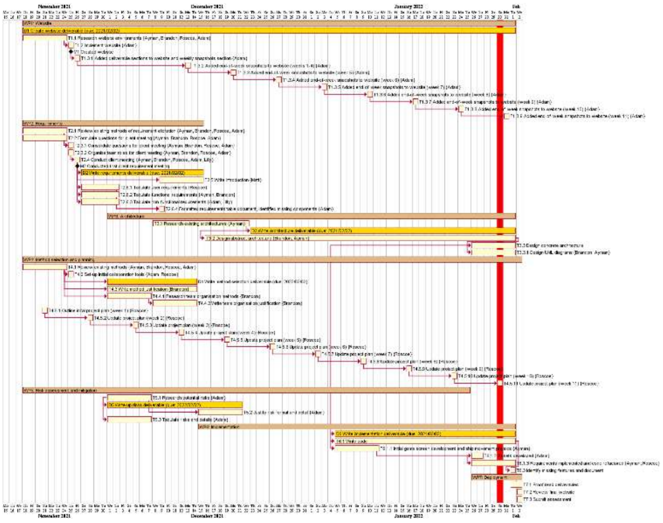


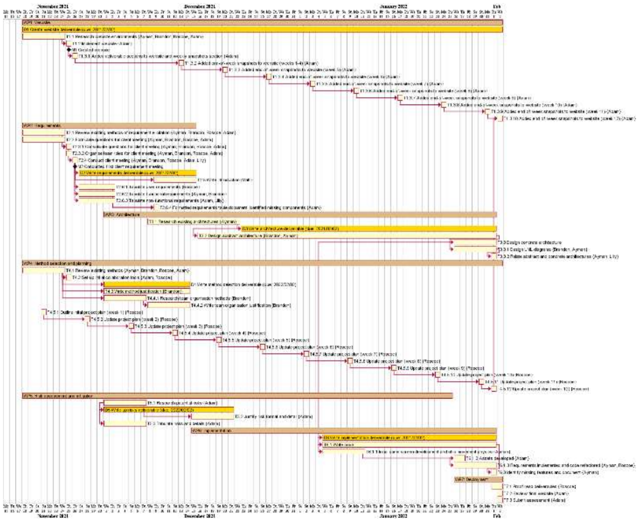
By the beginning of December, the plan stayed relatively consistent whilst also considering holidays, which may have disrupted the workflow slightly due to exterior commitments. Regular team meetings occurred throughout this period to keep everyone on the team informed about the current state of the project and change plans as necessary.











Towards the last week before the deadline, the plan changed to effectively making sure everything was ready by completing the code and accompanying documents. Everything was reviewed to ensure this was the case.