Method Selection and Planning

Cohort 1, Group 5
Team Name:
JAzZ MoLeS
Group Members:
Sophia Taylor, Lucy Wood, Mitchell Gilbert
Jamie Creed, Archie Adams, Zayed Iqbal

Software engineering methods: The software engineering method our group has decided to go forward with is the SCRUM agile engineering method. It's inclusion makes sense for our project as it emphasises iterative and incremental development, it allows us to focus on delivering a working software in short cycles that are called sprints.

Scrum:

- Setup
- Planning
- Implementation
- Review
- Deployment

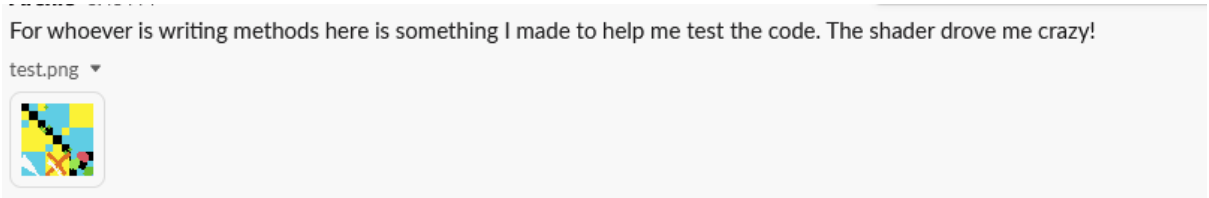
Justification:

- Quick response to feedback: Through continuous feedback loops (sprint reviews) the team is able to adapt features and mechanics based on in house testing
- Self-organising teams: Scrum allows each team to manage their work efficiently allowing each section to worked on without requiring the other to be done usually
- Communication: Sprint reviews which are meetings held weekly allows the communication between team members to be clear

Collaboration tools:

Slack:

- Real-time communication: Allows the team to quickly share information, and allow for quick discussions



Separate channels: By having the ability to add channels for each stage of the development process it allows for everyone to see the process of the team members and updates everyone on where we currently and allows meetings to be planned around it
Google Drive:

- Cloud storage: it provides a central location to allow for storing and accessing project files
- Document sharing and collaboration: It allows for real-time collaborative editing of documents like design specifications, user stories and project plans

Git:

- Version control: It allows for tracking changes and enables us to revert to previous versions if needed and collaboration.
- Branching and merging: It enables parallel development and enables efficient integration of different features

Justification for selected tools:
The tools we've chosen are all complemented with the scrum methodology:

- Slack: Supports the daily communication and collaboration needed during sprints.
- Git: Enables effective version controlling and collaboration on the code and is crucial for iterative development
- Google Drive: Allows a central location for files to be stored allows the management of project documents aligned with scrum practices

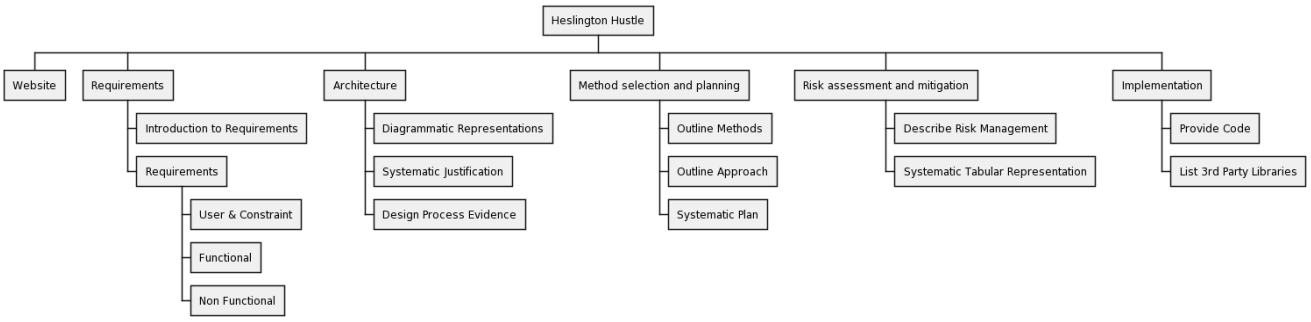
Alternatives Considered:
Project Management Tools: So while there are other tools like discord and jira. What slacks provides is the ability to have dedicated channels and threads which fosters focused communication and information organisation and allows threads to enable keeping discussion on specific topics within a channel preventing from information overload and maintaining a clear flow of conversation

Team Organisation

At the start of the process, we decided to allocate roles to each team member. We felt this would be beneficial as it would allow each member to best utilise their strengths. These were the roles we chose:

- Secretary
    - Lucy Wood
- Librarian
    - Jamie Creed
- Software Architect
    - Zayed Iqbal
- Scrum Master
    - Sophia Taylor
- Software Developer
    - Mitchell Gilbert and Archie Adams

After the first few weeks, we felt like these roles weren't working for us as a team. We chose to move to more fluid roles where people collaborated on tasks. This was a much better fit for us, and we used logbooks for each meeting to allocate tasks to people.
We felt the initial roles hindered the development of the project as they were too restrictive and forced people to only focus on one section of the project, rather than collaborating - this was a risk that we identified in the risk register (R1,R2).

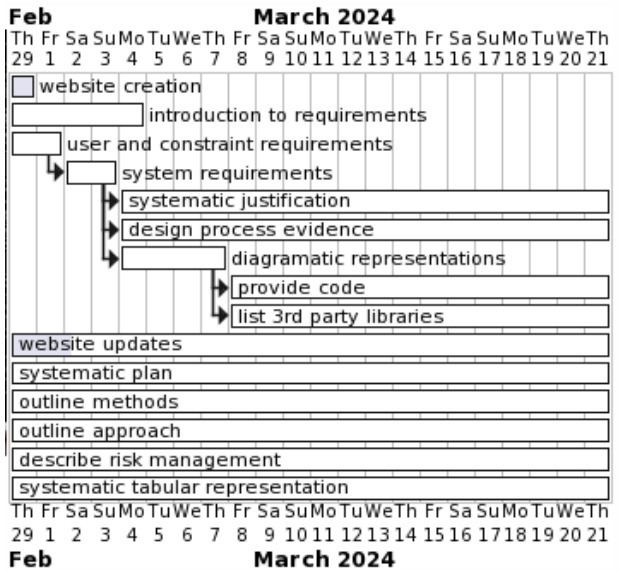


Deciphering tasks: We first went through the entirety of the product brief, as well as the assessment and made a checklist of all tasks that needed to be done. From there, we were able to determine the main sections and tasks that make them up. Once this was done, a breakdown diagram was made in plantuml as shown above.

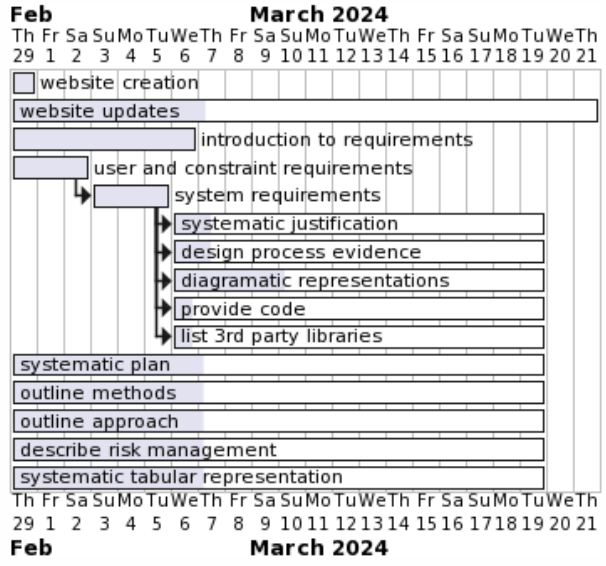
Task	*Person/People*
website	**lucy**
requirements	**sophia, archie, mitch, jamie**
introduction to requirements	jamie
user and constraint requirements	mitch + archie
system requirements (functional)	sophia
system requirements (non-functional)	sophia
architecture	**jamie, zayed**
diagramatic representations	zayad
systematic justification	zayad
design process evidence	jamie
method and selection planning	**zayed, sophia**
outline methods	zayad
outline approach	zayad
systematic plan	sophia
risk assessment and mitigation	**lucy**
describe risk management	lucy
systematic tabular representation	lucy
implementation	**archie, mitch**
provide code	mitch + archie
list 3rd party libraries	mitch + archie

Allocation of tasks: After decomposing the assessment and deciphering the necessary tasks, we then talked between ourselves and allocated ourselves for the rolls we felt most comfortable with. On top of this we tried to make sure that the workload and its specified marks were evenly distributed between everyone.

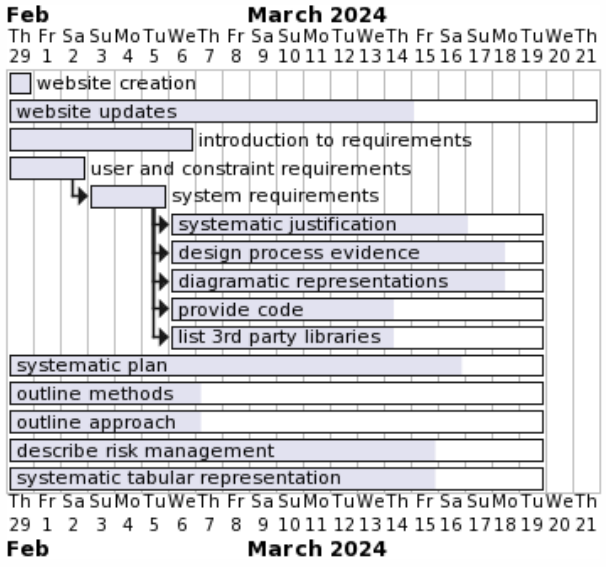
Now that the tasks were defined and had been allocated to team members, it was time to plan out our time through gantt charts - which are a visual representation of the prioritisation and dependencies of these defined tasks.



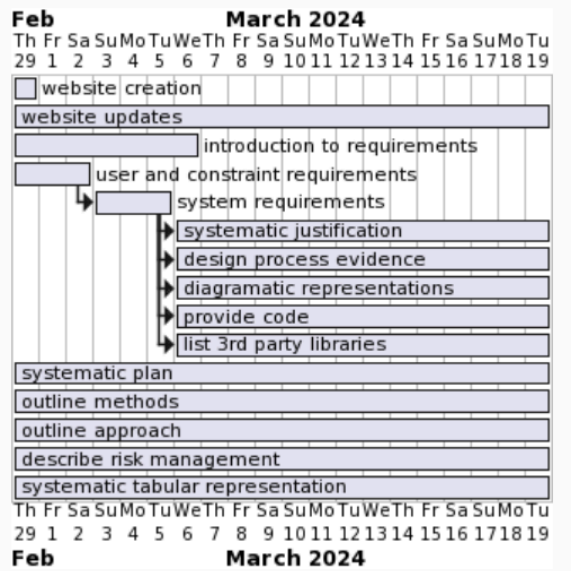
Week 1: During this first week we all sat down and looked through the breakdown diagram and allocation table. Using this paired with the assessment document, we were able to define what the dependencies were and the order in which it seemed best and most efficient to do things.



Week 2: This week we talked about our progress, and re-evaluated the dependencies of our allocated tasks. Through this, we decided that the diagrammatic representations would be better to be made alongside the code in case the code structure were to change, then those team members would talk amongst themselves to make sure they are accurate. The progress of the project as a whole is represented through the shading of the gantt chart.



Week 3: This week was simply a check in for everyone to see where we were at with our allocated tasks, and this progress was represented through the shading on the progress bar. In terms of the outlining methods and approach, now that we have developed further into the project, it should be written with more ease and efficiency. Overall everything looks like it will be finished on target ready to be proofread before submission.



Week 4:. We got all the individual components ready for Tuesday so we could sit down and proofread and potentially submit. The deadline for website updates was also changed to the potential new submission date to keep in line with the rest of the work done.

These gantt charts were made every Thursday for the week except for the last which was made on the Tuesday before the submission deadline date. And the group's overall progress is represented by the steady shading of the separate tasks.