

# Method Selection and Planning

[Note: This is not the document passed to us by Mallard, but instead a document from Smew's last assessment (changes in blue)]

## Method Selection

This project will make use of a structure similar to the waterfall model<sup>1</sup>. This ensures that large tasks are done in a logical order. The waterfall model suggests a testing phase, this will not be present within this project, instead testing will be a continuous task during the implementation phase using test driven development. The waterfall model will only be used loosely, to maintain structure, the main organisation aid to be used will be Scrum.

Scrum is a framework designed to be used by smaller development teams, in order for them to be able to tackle more complex and intimidating problems<sup>2</sup>. The frameworks creators recommend teams containing between 3 and 9 members. The team size and scale of the project make Scrum appropriate for our purposes.

Within scrum there are 3 major roles/groups:

- **Product owner**  
The product owner's primary role is to maintain the project backlog (pg. 2).
- **Scrum master**  
The scrum master ensures that the scrum framework is being followed correctly.
- **Development team**  
The team responsible for completing the items in the project backlog. Within this team there is no hierarchy.

For the purposes of this project, the Scrum master and product owner roles will be combined. This is due to the number of other commitments within the team, a combined role would ensure a more organised team environment. All those in the group will be members of the development team.

Within the scrum framework, two documents are used:

- **Project backlog**  
The project backlog contains all tasks to be completed throughout the project, each task should have a clear description.
- **Sprint backlog**  
The sprint backlog contains tasks to be completed in the current sprint.

Both backlogs will be used in this project, with the sprint backlog being implemented using GitHub's issues feature.

Scrum recommends daily meetings. This would not work within the bounds of the project. Instead we have opted to have 3 meetings per week, including one long collaborative session. Using this, we can have multiple sprints per week, and stay in close contact.

Throughout the development process, Test Driven Development (TDD) will be used. TDD entails running tests every time a new procedure is implemented. All tests will be documented and saved so they can be rerun if code is updated. With this method we can ensure that all code is tested fully, minimising the risk of bugs and therefore a poor final product.

---

<sup>1</sup> ISTQB Exam Certification. "What is Waterfall model- advantages, disadvantages and when to use it?", [istqbexamcertification.com](http://istqbexamcertification.com). [Online] Available: <http://istqbexamcertification.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it/> [Accessed: 15 October 2015]

<sup>2</sup> K. Schwaber and J. Sutherland, *The Definitive Guide to Scrum: The Rules of the Game*. ScrumInc., 2014. [E-book] Available: <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-US.pdf#zoom=100> [Accessed: 15 October 2015]

Throughout the project, changes to previous sections can be made. For example, during development the requirements may change. As a team we intend to be flexible in cases like this, in order to increase the quality of the final product produced.

## Team Roles

The Scrum framework requires a non-hierarchical development team, with no specific roles. For this reason, throughout the project, the team will have a fluid structure, with only one major role, the combined Scrum master and product owner.

### Individual Roles

Scrum Master - Will Strachan

Product Owner - Will Strachan

Development Team - Avinash Dwarapu, James Harrison, Timothy Ashton, Ranjit Bassi, Nathan Carter

For each segment (release) of documentation, one person will be assigned the role of overseeing the production of formal write-ups. This prevents inconsistencies within the documents, as well as ensuring the same writing style is used throughout.

Throughout the project, delegation of tasks will be required. In order to minimise risks (discussed in the 'Risks and Mitigation' section), at least two members of the development team will be assigned to a task. This prevents absence halting progress, with at least one person still being able to complete the task should a team member be unable to continue.

During the implementation, process the development team will be split into two groups each with 3 members. One group will focus mostly in implementing the game, whereas the other will focus on producing documentation. Both groups will maintain contact to ensure that produced work is consistent, and all members of the development team have an understanding of all work being produced.

#### Group 1 (Code Implementation)

- Timothy Ashton
- Avinash Dwarapu
- James Harrison

#### Group 2 (Documentation)

- Will Strachan
- Nathan Carter
- Ranjit Bassi

## Development and Collaboration Tools

### - Google Drive

Documentation will be stored on Google Drive so that they are available to all team members for viewing and editing simultaneously.

### - Git

Version control will be handled by git. The code will be hosted on Github, allowing for easier collaboration. Github provides additional features that would be helpful to our development lifecycle such as an issue tracker for managing items in the project backlog and assigning them to team members and a milestone tracker for managing sprints and release deadlines.

### - Facebook Messenger

For communication and coordinating meetings, we decided to use Facebook chat, since all of the team members are available on the platform and check it regularly.

### - Skype

Outside of term time, some members of the development team will not be available for the group coding sessions. During these periods, we will use skype to maintain live contact.

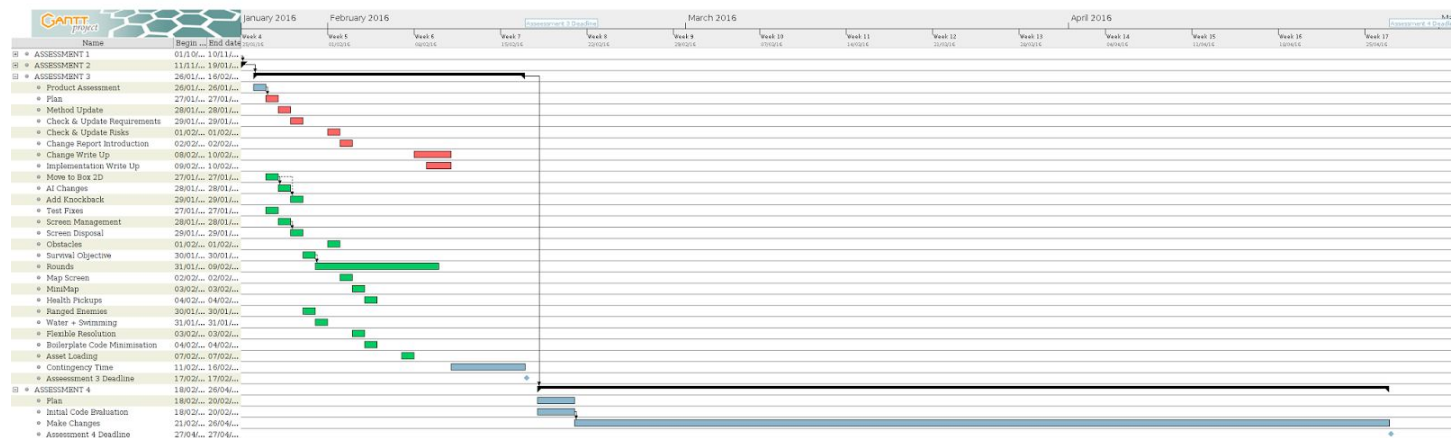
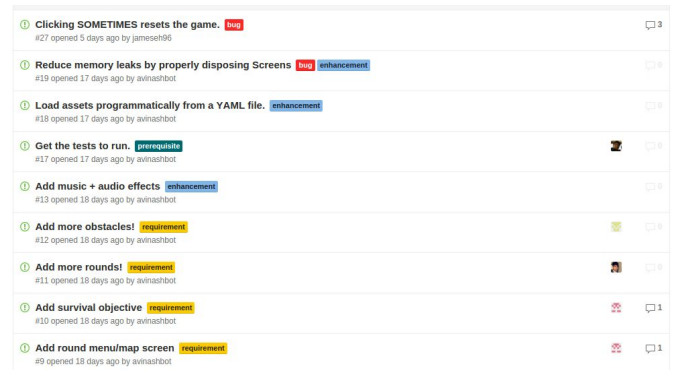
## Planning

In order to complete the third assessment, a plan had to be created. Initially, each section of documentation had to be broken into its constituent tasks, and their individual timings. The tasks were then ordered and organised into a Gantt chart. Dependencies and potential overlaps were decided and implemented into the plan.

After evaluating the previous team's work, we put together to lists of tasks, one for documentation, one for implementation. Both of which are available of the website (documentation [here](#), and implementation [here](#)).

Task ID	Task Name	Responsible	Done?	Document	Description
#1	Reading	WS	Y	All	Read through all documentation, analysing for tasks
#2	Severity/Likelihood Update	WS	Y	Risk + Mit. Assess.	Split severity and likelihood into separate columns
#3	Intro update (comp. #2)	WS	Y	Risk + Mit. Assess.	Update introduction to cope with task #2
#4	GUI Changes		N	GUI Report	Update to include changes made throughout implementation
#5	SCRUM Methodology	WS	Y	Method and Plan	Update team structure according to SCRUM Methodology
#6	Plan	WS	Y	Method and Plan	Plan for assessment
#7	Check requirements	WS	Y	Requirements	Read requirements in order to advise implementation team
#8	Document test changes	NC	N	Testing	Document changes made to tests in order to make them run
#9	Update manual		N	User Manual	Update user manual to cover new features added
#10	Design Decisions		N	Implementation	Update Implementation to cover new design decisions
#11	Summarise change management	NC	Y	Change Report	Write introduction to change document to summarise how changes were monitored
#12	Coallate Changes		N	Change Report	List changes made throughout assessment
#13	Write up change report		N	Change Report	Write up list produced in #11, with justification

- The documentation task list was created using a spreadsheet. Within it tasks are listed, each with a description, the person responsible, and it's progress (done? Y/N).
- The implementation task list was created using GitHub issues. This allowed us to reference commits when resolving issues, as well as who resolved the issue, keeping an air of traceability.



The Gantt chart was produced with the tasks decided using the lists described above, with consideration for task dependencies. A sample of the Gantt chart can be seen above, with a higher resolution version being available on the website [here](#). The Gantt chart follows the entire time frame of each assessment, with specific tasks assigned to periods of time, having taken team availability and dependencies into consideration. A piece of software called GanttProject was used this time to create the Gantt chart as dates can be used to generate bars on the chart and so it allowed for easier chart modification.