Method Selection and Planning

4.1 Chosen Methods

SEPR is a small team project which must be completed within a relatively short period of time. For this reason, we thought it would be most suited to the **Agile** approach. We have also decided to incorporate more traditional approaches to take advantage of the benefits they offer.

We have selected the detailed requirements phase & documentation from the Rational Unified Process (RUP) [1], while keeping the Agile performance and adaptability that Scrum has to offer. This is important because the requirements are volatile and subject to change, and using an Agile process helps to backtrack through changes in ways that traditional methods don't.

RUP is effective for requirements and documentation because separates each iteration of the development process into nine distinct disciplines, and the system life cycle into four phases. These phases are:

- Inception
- Elaboration
- Construction
- Transition

The RUP framework contains very clearly defined measurement criteria for all phases, utilising Use Cases and Prototypes to ensure the product is developed efficiently to the requirements set in the Inception Phase.

Scrum [2] is perhaps the most popular agile development framework, and is focussed on short iterations of the development cycle enabling the customer to receive some part of their product in the shortest possible time. Our project plan [3] is divided into Sprints as per Scrum's structure, which breaks up the backlog of work into manageable chunks and enables the the team to work on multiple sprints concurrently. An important side effect of these short rotations is that a complete change of requirements will usually only set the team back by one sprint, as the customer will have given feedback on the previous release. We would use Scrum ideas to plan out all steps with fixed sprint periods for each of the main focusses of the game (GUI, Conversation, Clues etc.) and will make use of these standups to discuss the developments that have occurred and, perhaps more importantly, the problems that we are individually facing. Another advantage of scrum is that it can adapt well to our customer's changing requirements unlike many other frameworks.

Extreme Programming [4] promotes an ethos to build and test parts of code quickly and efficiently, which would be useful to incorporate into our project due to the strict time constraints. As there is a strong emphasis on building unit tests for all code, our testing would be concise and organised, with a reduced risk of unexpected errors or bugs. Where possible we will borrow the approach of paired coding from Extreme Programming to provide us with a more accurate codebase in the first instance; hopefully reduce the amount of time bug-hunting. However, because each team member has different commitments and an irregular schedule, this may be difficult to exploit consistently. To avoid this risk, we will complete most of our paired programming within the scheduled SEPR slots on our timetables.

4.2 Collaboration Tools

With respect to development tools, we have decided to use **Google Drive** [5] for online collaboration on documents, and this is where we will host our project plan which we will actively keep updated. Google Drive is so useful because not only are all the documents available remotely at all times, it gives us the option to suggest edits to certain files without changing them directly if we so desire.

We used **Google Sheets** to make the Gantt chart. This was designed as a team, so that we could make sure we were all happy with our roles. This was used as the initial plan for our project. We uploaded this onto Google Drive so it was available for all members of the team to reference whenever it was required.

We will use **Git** to keep track of source code & assets as it is an industry trusted way of making safe changes to important files. It will become very useful for bug fixing and implementing separate modules as branches before making commits to the master. We will be using a **GitHub** [6] repository to store the aforementioned files as this is an online, distributed storage space that will allow us to program remotely with no chance of data loss. Some members of our team are more confident with using Git than others, although we have all attended introductory lectures and are at least confident with the principle ideas.

We communicate daily using the **Facebook Messenger** [7] platform to arrange face to face meetings as well as discuss project work when we are working remotely, purely because everyone is familiar with the platform and the mobile notifications are incredibly beneficial for when members are not at a computer.

We have used **Microsoft's Visio** [8] to design the UML [9] and other such diagrams as the presets and built in templates allow for a fast and efficient workflow.

We will use the **IntelliJ Java IDE** [10] to develop our code. We have chosen to use this IDE because it is one of the best available, is free with a student license, and one member of our group is familiar with some of its more useful features, for example, automattic Getter/Setter methods.

4.3 Organisational Methods

For the initial planning phase of the project, we worked together as a team to gather the overall requirements so we could all agree on the basic plan of the game. We then decided it would be best to split off into small groups. This was so we could maximise efficiency by working on different parts of the project at the same time. Each section was assigned a team leader, who was responsible for ensuring that particular part of the project would be completed on schedule to a high quality.

Despite having roles within the small sub-teams we were working in, we also had overall roles within the project. We decided our team roles based on our strengths and weaknesses. It was also important to ensure that everyone was happy with their role, as this meant that work would be completed to a high standard. We assigned our team roles based on the Belbin Team Roles [11].

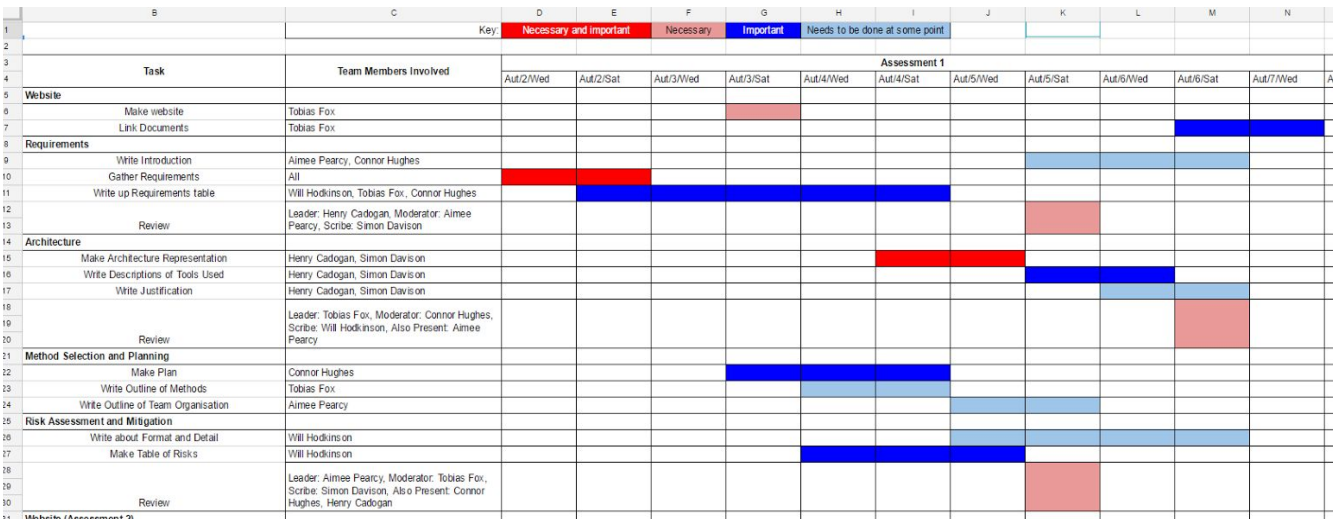
Our overall team roles for the project are:
Completer Finisher: Henry Cadogan - polishes and scrutinises end product for errors.
Co-ordinator: Connor Hughes - focuses on objectives, delegates team roles.
Implementer: Tobias Fox - plans a strategy and ensures it is carried out.
Monitor Evaluator: Will Hodkinson - weighs up pros and cons impartially.
Plant: Simon Davidson - creative problem solver.
Shaper: Aimee Pearcy - provides drive to ensure team does not lose motivation or get sidetracked.

Our timetabled slots are mainly used for discussing architecture and requirements as a group with other team members. We also use this time to ask lecturers any questions we might have about the brief, and also to ask our customer any questions we had about specific parts of the project. Outside of our timetabled slots, we meet once per week on a Friday morning for a Scrum/ standup. During this period, we quickly update each other on what we've accomplished during the past week, what problems we are experiencing, and set ourselves new tasks for the following week. We then work separately in week long *Sprints* to complete the work we'd set ourselves. If we have any questions during this time, we could contact one another via Facebook messenger.

We decided that this would be the most effective way to organise our meetings because our project is relatively short, and it was important to get things done quickly. Sprints also helped to motivate our team members, because it meant that roles were changed frequently and everyone got to give their input on several different parts of the project. Towards the end of the assessment we started having reviews of each of the main documents to ensure that everybody in the team was confident of the quality of each of the individual parts.

4.4 Plan

The plan has been done as a Gantt chart with each column representing half a week (being identified by when the half week ends) and each row representing a task to be done. One column shows which team members are primarily involved in each task, to minimise confusion and to prevent members getting too involved with other sections of the project. The plan can be changed at review sessions if certain combinations of team members aren't suited to work together and if deadlines need to be changed. Tasks were assigned to team members based on any experience they had with the specific task domain, and if the same task is returned to later in another assessment (for example in the change report), the same team member(s) that were involved in the previous section will be assigned again to maximise productivity. The red tasks are prioritised (defining the critical path for the project), and must be completed before the blue tasks can be worked on due to dependences, while deeper colours signify more important tasks.



		Key:	Necessary and important		Necessary	Important	Needs to be done at some point					
	Task	Team Members Involved	Assessment 1									
			Aut/2/Wed	Aut/2/Sat	Aut/3/Wed	Aut/3/Sat	Aut/4/Wed	Aut/4/Sat	Aut/5/Wed	Aut/5/Sat	Aut/6/Wed	Aut/6/Sat	Aut/7/Wed
	Website												
	Make website	Tobias Fox											
	Link Documents	Tobias Fox											
	Requirements												
	Write Introduction	Aimee Pearcy, Connor Hughes											
	Gather Requirements	All											
	Write up Requirements table	Will Hodkinson, Tobias Fox, Connor Hughes											
	Review	Leader: Henry Cadogan, Moderator: Aimee Pearcy, Scribe: Simon Davison											
	Architecture												
	Make Architecture Representation	Henry Cadogan, Simon Davison											
	Write Descriptions of Tools Used	Henry Cadogan, Simon Davison											
	Write Justification	Henry Cadogan, Simon Davison											
	Review	Leader: Tobias Fox, Moderator: Connor Hughes, Scribe: Will Hodkinson, Also Present: Aimee Pearcy											
	Method Selection and Planning												
	Make Plan	Connor Hughes											
	Write Outline of Methods	Tobias Fox											
	Write Outline of Team Organisation	Aimee Pearcy											
	Risk Assessment and Mitigation												
	Write about Format and Detail	Will Hodkinson											
	Make Table of Risks	Will Hodkinson											
	Review	Leader: Aimee Pearcy, Moderator: Tobias Fox, Scribe: Simon Davison, Also Present: Connor Hughes, Henry Cadogan											

Figure 1. Screenshot of part of the plan between the dates of Assessment 1 (The full, detailed plan can be found on our website https://henrycadogan.github.io/Clued-up/ass1/webfiles/ProjectPlan.pdf).

4.5 Bibliography

[1] S. Jacobson, “The Rational Objectory Process - A UML-based Software Engineering Process”, *Rational Software Scandinavia AB,* 2002. [Online]. Available: http://www.iscn.at/select_newspaper/object/rational.html [Accessed: 9- Nov- 2016].

[2] “Learn About Scrum", *Scrum Alliance*, 2016. [Online]. Available: https://www.scrumalliance.org/why-scrum [Accessed: 30- Oct- 2016].

[3] H. Cadogan, S. Davison, T. Fox, W. Hodkinson, C. Hughes and A. Pearcy, "Plan1", *Wedunnit!*, 2016. [Online]. Available: https://henrycadogan.github.io/Clued-up/ass1/webfiles/ProjectPlan.pdf [Accessed: 8- Nov- 2016].

[4] "Extreme Programming", *Cs.usfca.edu*, 2016. [Online]. Available: http://www.cs.usfca.edu/~parrt/course/601/lectures/xp.html . [Accessed: 30- Oct- 2016].

[5] *Drive.google.com*, 2016. [Online]. Available: http://drive.google.com/ . [Accessed: 8- Nov- 2016].

[6] "Build software better, together", *GitHub*, 2016. [Online]. Available: https://github.com/ [Accessed: 8- Nov- 2016].

[7] *Facebook*, 2016. [Online]. Available: https://www.facebook.com/ [Accessed: 8- Nov- 2016].

[8] "Visio 2016 | Professional Flow Chart & Diagram Software". *Products.office.com*. N.p., 2016. Web. 19 Oct. 2016. [Online] Available: https://products.office.com/en-gb/visio/flowchart-software [Accessed: 19- Oct- 2016]

[9] H. Cadogan, S. Davison, T. Fox, W. Hodkinson, C. Hughes and A. Pearcy, "ClassDiagram", Wedunnit!, 2016. [Online]. Available: https://henrycadogan.github.io/Clued-up/ass1/webfiles/ClassDiagram.pdf [Accessed: 8- Nov- 2016].

[10] "IntelliJ IDEA the Java IDE", *JetBrains*, 2016. [Online]. Available: https://www.jetbrains.com/idea/ . [Accessed: 01- Nov- 2016].

[11] "Belbin Team Roles | Belbin", *Belbin.com*, 2016. [Online]. Available: http://www.belbin.com/about/belbin-team-roles/ [Accessed: 07- Nov- 2016].