1.0 Introduction

1.1 Outline

The development team for this project consists of six individuals, and we believe such a small team can only benefit from using an agile based method. The use of other models, such as waterfall or spiral, are reliant on large numbers of staff and resources. It is these models that are (for the most part) inflexible, which given the nature of this project is a concern. We have therefore decided to use a SCRUM based method with the inclusion of characteristics from Extreme Programming (XP). Much of the method decision and design process was inspired by Laurie Williams 'Risk Management' [7].

Our SCRUM development cycles are iterative and flexible. Each cycle (sprint) has a prioritized list of deliverables (a backlog) with an estimated timeline. Members of the team will form small groups (a pack) and work together to complete elements of the backlog, often working in pairs as described in the XP method. Each pack is advised to have a regular meeting (a scrum) to discuss their progress and report any issues that may arise. If these issues prove to be larger than the pack can handle alone, they report the issues to the whole team either through our various online communication methods or in the scrum of scrums - a meeting where all of the pack's are present to discuss overall progress and any rising issues.

1.2 Method Justification

On receiving the brief we researched relevant models that may be adopted for this project, one of which was the waterfall model. The nature of this model means that once requirements have been finalised it would be extremely difficult to adapt to any manner of change. This is a concern, as the customer may request the development of additional features or the adaption of present requirements. Such a request will likely result in compromise, while potentially leaving the customer unsatisfied and could lead to a large amount of stress and extra workload being placed on the team as we adapt accordingly. As such the use of a flexible agile method is best suited for this project.

The size and experience of our organization greatly reduces the number of viable models. Through consideration of Laurie Williams 'Risk Management' [7], we come to the decision that an agile method was best suited. The majority of the team while capable have little experience with software development, while the remainder has some manner of experience and are very capable. This along with our small team size supports the characteristics of agile development (that being small teams and high levels of teamwork) this is also what Laurie Williams [7] recommends in her paper for teams of a nature similar to ours.

The SCRUMs communication methods (regular scrums and scrums of scrums) allow for efficient team coordination and management, ensuring that each member is aware of the current state of a sprint and is free to comment on progress. This allows for effective problem solving, prioritization and as a means of motivation.

The choice to include characteristics from XP was to address two main concerns. The primary lies with individual skill proficiency as SCRUM requires that all members of the team are equally skilled in all areas. This allows for anyone to continue a piece of work whereas we as a team have varying strengths and weaknesses hence the inclusion of XP elements. Secondly, that even with the use of a product owner there's a concern that we may still develop incomplete requirements or misinterpret the client's mental model. This would prove to be a both a waste of resources and a failure in our obligations to the client.

To overcome these problems we have adopted pair programming and client participation from XP. We've adapted pair programming so that an experienced programmer and an inexperienced programmer will work side by side (as opposed to two to a work station). This will not only support the inexperienced members and improve their skills, but also the the two members may have a different approach to problem solving - adding another dimension to our work. Client participation is paramount to the development of a suitable product, therefore we have decided that communication with the client should be clear and frequent during all stages of the project in order to meet their requirements.

1.3 Tools

To allow for efficient and effective development and to improve the overall management of the project we've utilised many tools, which are detailed below. These tools have helped in many areas, including documentation, planning, communication and code management.

Version control is conducted through the use of Git through a GitHub repository. This allows for members to access/modify code from any machine before uploading it to a remote repository, therefore reducing the possibility of lost work. In addition to this, the use of branching will help different pack's (groups) implement different sections of the game at the same time without having to worry about the effect this will have on other groups work. Once these individual features have been implemented by each pack, they can be merged back to a central point. Using branches also ensures the master version of the project won't be modified until the most recent additions/revisions have been thoroughly tested, and as such we predict less issues with merging different pack's code.

Document creation and management is conducted through the use of Google Drive. This allows for all members to monitor development, peer review and edit files simultaneously. Google Drive also allows for the sharing of documents, this provides the opportunity to share relevant documentation with stakeholders and the customer - allowing for effective communication. Another feature is that of continuous file saving thus ensuring that no work is lost.

Planning is conducted in Google Calendar and Google Charts. Google Calendar allows for the creation of a shared calendar and is therefore used as a means of organising scrums, team meetings and recording various deadlines. The project plan is created through the use of Google Charts' Gantt Chart and is hosted on the team website for all to see. This chart visually represents our progress, showing all of our tasks for the current assessment along with our own internal deadlines.

Communication is conducted through the online platform Slack. This tool allows for communication over various channels such that no member receives irrelivant information. Slack also allows for integration of other tools, such as Google Calendar and Github, meaning all members will receive notifications of any changes to planning or code implementation. When pack's are unable to meet face-to-face for a scrum, Skype is used to conduct these meetings, ensuring that any geographical displacement that may occur will have a minimal effect on development.

The Unity game engine will be used to support our development. This engine comes with its own GUI editor, which allows the game's visuals to be changed quickly and easily, as well as any game object properties. Unity also provides a framework that allows us to just focus on writing game logic, as it handles more complex tasks such as rendering, allowing the team more time to develop actual game features.

2.0 Team Organisation

Task assignment is conducted at the start of a sprint, along with task analysis, where members discuss the skills/traits required for the completion of each task. Following the task analysis members are encouraged to consider their own strength and weaknesses, and then volunteer for suitable roles. Should there be a conflict (such as too many volunteers or a lack of volunteers) the scrum master will delegate the role. To help with this delegation all members of the team have completed a personality test. We believe that this method allows for members to work on tasks that they're suited for. In addition, having a clear hierarchy ensures that conflicts are always dealt with quickly and consistently.

Below are a list of continuing responsibilities for all members of the team. The abbreviations refer to our personality type as identified by a Myers-Briggs/Jungian model [1]. You can find a description of the job roles on our website here [6].

Theo Barber-Bany
Scrum Master [6]
(ENTJ) [8]

Theo is well suited to managerial roles as he is an efficient worker and energetic. Theo is also diligent as such he's constantly monitoring progress and ensuring that any problems are resolved quickly.

Julia Fellows
Secretary / Product Owner [6]
(INFJ) [2]

Julia is both insightful and a great communicator. She often understands the heart of a matter therefore she understands our clients needs and is able to communicate this effectively to the team.

Tim Bradgate
Librarian / Product Owner [6]
(INFP) [3]

Tim is open-minded and creative, meaning he's suited to developing the customer's vision. He is also very practical, meaning the customer is often informed of the difficulties associated with their requirements.

Matthew Dunn
Report Editor [6]
(ISFJ) [4]

Matthew is reliable and supportive. This allows for him to asses documentation and provide constructive criticism quickly. He's also logical allowing him to edit documentation so that it's presented concisely.

Matthew Turton Parry
Report Editor [6]
(ISFJ) [4]

Matthew is enthusiastic and observant. This allows him to edit documentation to ensure a persistent narrative without losing motivation. Matt is also very practical as such he excels at explaining technical matters succinctly

Mikolaj Bernaciak
Risk Manager [6]
(ESTP) [5]

Mikolaj is rational, perceptive and direct meaning he's effective at prioritization and mitigation of risks. In addition to this his direct nature allows him to tackle the heart of the problem and develop effective mitigation.

During the first phase Julia and Tim worked on requirement elicitation, analysis and documentation, as this matched their assigned roles of product owners. Architecture was completed as a team seeing as during implementation all members are likely to be coding. Matthew Dunn and Matthew Turton Parry were responsible for the documentation of our chosen methods, team organization and the systematic plan which is due to their observant and patient natures. The initial risk documentation and analysis was done by Theo and Mikolaj as they're both perceptive and efficient.

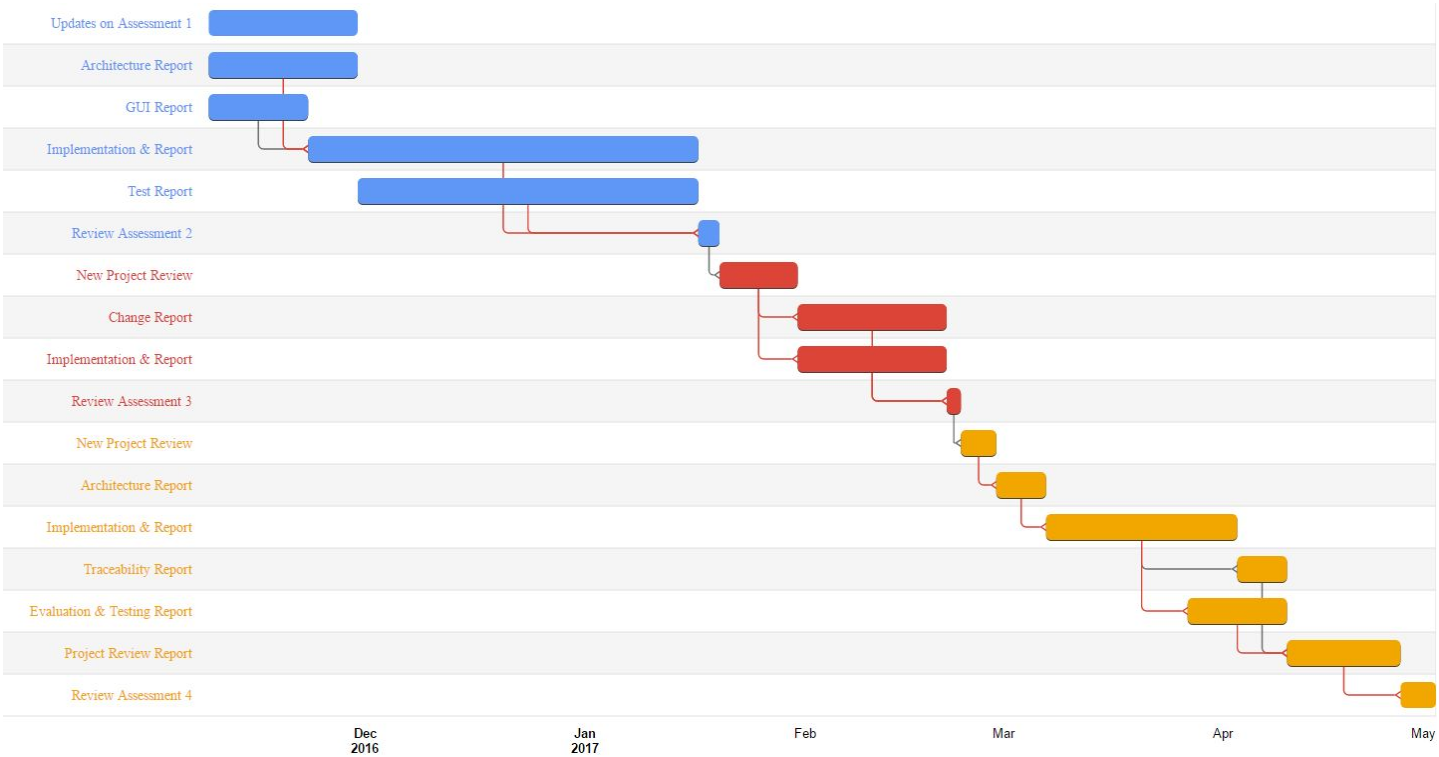
3.0 Systematic Plan

We have decided to record our plans in detail through the use of a Gantt chart, which is accessible through our website. This provides a simple and effective way of displaying the progress of our work along with important details, such as their estimated durations and their dependencies. One issue that we have encountered with the integration of the chart is the lack of specific start and end dates. This is overcome by including an online backlog where tasks are discussed in detail (including their duration) for each development cycle.

To develop a systematic plan for the remainder of the project, we first considered the project structure/timeline provided by our managers (the SEPR project brief). We decided to keep the Gantt chart simple and maintain a more detailed backlog elsewhere - this was done to encapsulate information and declutter the chart, making it easier for the team and client to process. Estimation of task duration was conducted during a team discussion, where we considered past tasks and compared their difficulty to upcoming tasks. To ensure that our lack of experience had minimal effect on our timings (i.e. not assigning enough time for each task) we then added some extra time to the estimate .

Starting dates were decided based on the delivery date of each assignment, with consideration for each tasks estimated completion time. We have ensured that all tasks are finished early to allow for a period of peer review. This also gives us some leeway to complete tasks that may overrun due to unforeseen circumstances.

3.1 The Gantt Chart



In the above image, development cycles for each assessment are grouped by colour, such that; blue is assessment two, red is assessment three and orange is assessment four. The task dependencies are represented by a solid black line between tasks, while the critical path delay for each assessment is represented by a solid red line.

Each task in assessment two has it's own backlog where it's been subdivided into smaller tasks that need to be accomplished, these are stored along with a description and a time estimate. This backlog can be accessed here [9], note that tasks are listed in order where the following tasks are dependant on its predecessors.

Bibliography

[1] 16 Personalities, "Our Theory," [Online]. Available: https://www.16personalities.com/articles/our-theory [Accessed 4 November 2016].

[2] 16 Personalities, "INFJ Personality," [Online]. Available: https://www.16personalities.com/infj-personality [Accessed 4 November 2016].

[3] 16 Personalities, "INFP Personality," [Online]. Available: https://www.16personalities.com/infp-personality. [Accessed 4 November 2016].

[4] 16 Personalities, "ISFJ Personality," [Online]. Available: https://www.16personalities.com/isfj-personality. [Accessed 4 November 2016].

[5] 16 Personalities, "ESTP Personality," [Online]. Available: https://www.16personalities.com/estp-personality [Accessed 4 November 2016].

[6] SEPR, "Agile Team Roles," 4 November 2016. [Online]. Available: https://seprated.github.io/agile.html. [Accessed 4 November 2016].

[7] L. Williams, "Risk Manegment," 2004. [Online]. Available: http://agile.csc.ncsu.edu/SEMaterials/RiskManagement.pdf. [Accessed 4 November 2016].

[8] 16 Personalities, "ENTJ Personality," [Online]. Available: https://www.16personalities.com/entj-personality. [Accessed 4 November 2016].

[9] SEPR, “Backlog”, 6 November 2016. [Online]. Available: https://seprated.github.io/Assessment1/Backlog.pdf. [Accessed 6 November 2016].