

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, 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.

Implementation will be completed in a SCRUM-like fashion with one small alteration; on completion of a sprint (i.e. implementation of a feature or function) the developers will test this code. This approach is inspired by test driven development, this will increase development time but should reduce the number of issues caused by verification and integration.

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. 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 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.

Implementation is conducted through the use of what we have come to call Open Plan Programming (OPP) which is described here [6]. Through adopting this method we are able to support inexperienced members and as such are improving the quality of their delivered work. In addition this allows for more experienced members who are either dealing with complex issues, or are considering the best approach to solving a problem, to communicate with the team so that they may develop new and improved solutions. This also allows for all members to be aware of potential development issues and ensures that anyone may quickly discover how features are implemented when integrating them into their own work. Through the adoption of this method we hope to achieve a high standard of coding with innovative and simple solutions to complex problems.

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.

Version control is conducted through the use of Git through a GitHub repository [10]. 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 allow different pairs to implement different sections of the game at the same time without having to worry about the effect this will have on the work of others. 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 pieces of code. Git was chosen over other version control methods, such as SVN, due to the experience of the team. Many members are already familiar with the use of Git, as such version control mistakes are unlikely to occur - this could not be guaranteed in the use of SVN. Another attraction of using Git and GitHub over other platforms was the integration of ZenHub [16] which is a agile project management tool. This allows for easy task assignment, backlog management and quality control, meaning the team can develop in a structured manner in which they can easily visualize the current state of the project.

Document creation and management is conducted through the use of Google Drive [11]. 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. We considered other various document creation and management methods, the most promising of which was the use of version control and Microsoft products (i.e. Word, Excel etc.). After some discussion and research, the team were concerned that as members were developing on a variety of platforms compatibility issues may arise. Through using an online editor we will be able to avoid such issues.

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 [12] 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. The use of Google Calendar was greatly influenced by its integration into most communication platforms, this ensures that all members will be aware of any scheduled meetings/deadlines. A similar reasoning concerns the use of Google Charts, when deciding on monitoring progress Gantt charts were an intuitive choice and the use of Google Charts allows for this to be hosted on a website - providing easy access to any interested party.

Communication is conducted through the online platform Slack [13]. This tool allows for communication over various channels such that no member receives irrelevant 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 [14]. 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.

Testing will be conducted through Unity Test Tools and Travis [15]. Travis was chosen as it supplies an automatic method for running and produce test logs, through using travis we wish to ensure efficient and simple testing. It also provides notifications to the team on the results of this build ensuring that all are aware of arising issues.

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 / **Lead Developer** [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 / **Method Manager** [6]
([ISFJ](#)) [4]

Matthew is reliable and supportive. This allows for him to assess 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 / **Lead Developer** [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.

REMOVED PARAGRAPHS CAN NOW BE FOUND HERE [6]

2.1 Phase 3 Roles

The GUI design was completed by Theo and Tim as they have provided most input when attempting to generating effective and user friendly designs. The change documentation of which was undertaken by X as they have proven to be proficient in creation of clear and succinct documentation. Matthew Dunn, Julia and Mik worked respectively on updating methods, requirements and risks as this is encompassed in their individual; roles. Implementation of the project was conducted as a team to ensure that we successfully deliver a timely and quality product. Lastly testing followed the same procedure as phase 2, as this proved to be a simple and efficient system.

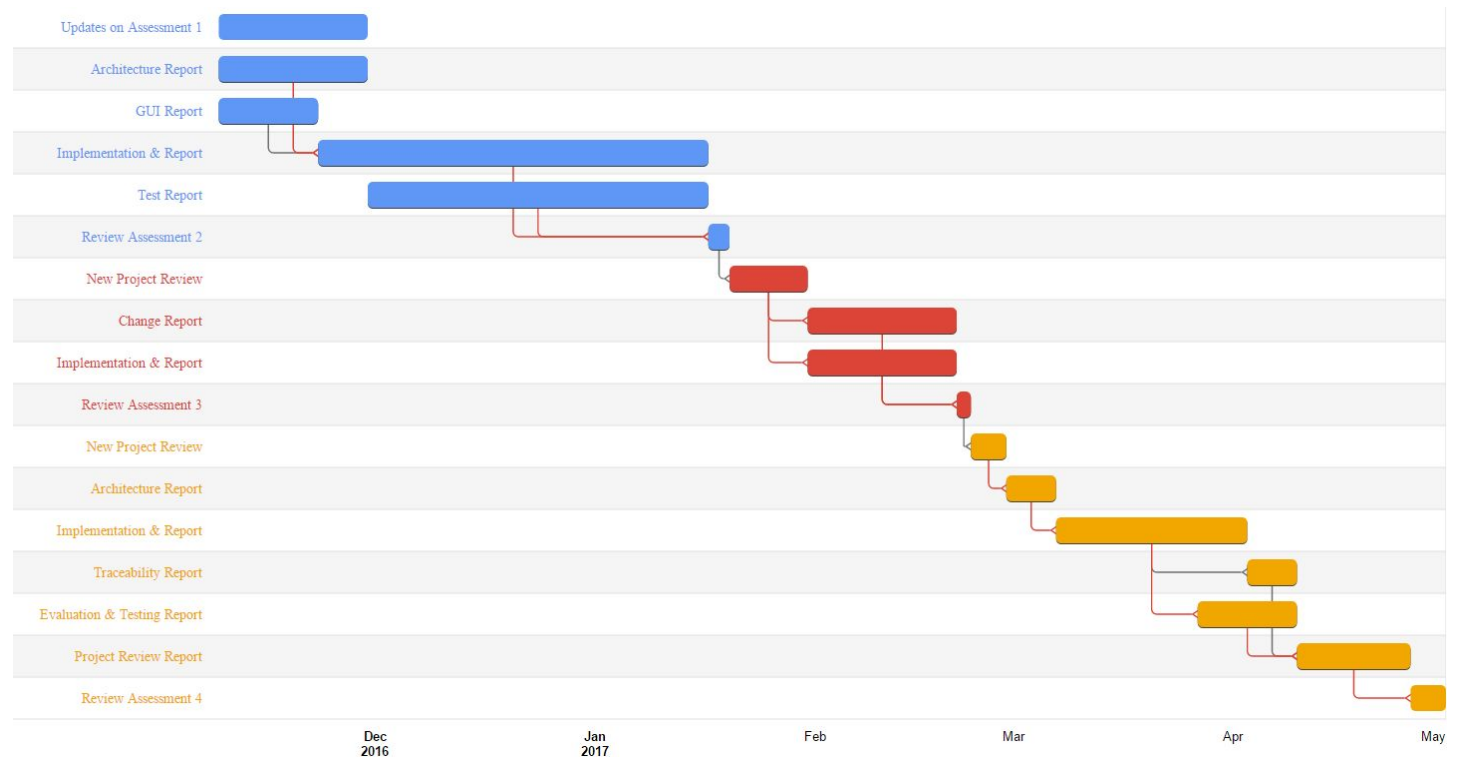
3.0 Systematic Plan

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. This can be found online [here](#) [18]. Each assessed task/process 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 and can be accessed by the following links.

Assessment 1: [Backlog](#) [9]

Assessment 2: [Detailed Gantt Chart with Task Dependencies](#) [19] / [Backlog](#) [20]

Assessment 3: [Detailed Gantt Chart with Task Dependencies](#) [21] / [Backlog](#) [22]

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].
- [10] GitHub, "How people build software," [Online]. Available: <https://github.com/>. [Accessed 18 January 2017]
- [11] Google, "Google Drive - Cloud Storage," [Online]. Available: <https://www.google.com/drive/>. [Accessed 18 January 2017]
- [12] Google, "Gantt Charts," [Online]. Available: <https://developers.google.com/chart/interactive/docs/gallery/ganttchart>. [Accessed 18 January 2017]
- [13] Slack, "Slack: Where work happens," [Online]. Available: <https://slack.com/>. [Accessed 18 January 2017]
- [14] Unity, "Unity - Game Engine," [Online]. <https://unity3d.com/>. [Accessed 18 January 2017]
- [15] Travis CI, "Travis CI - Test and Deploy with Confidence," [Online]. Available: <https://travis-ci.org/>. [Accessed 18 January 2017]
- [16] ZenHub, "ZenHub - Agile GitHub Project Management," [Online]. Available: <https://www.zenhub.com/>. [Accessed 18 January 2017]
- [17] Architecture design tool
- [18] SEPR, "SEP-R-ated," [Online]. Available: <https://seprated.github.io/>. [Accessed 18 January 2017]
- [19] SEPR, "Plan," [Online]. Available: <https://seprated.github.io/assessment3.html>. [Accessed 19 January 2017]
- [20] SEPR, "Backlog," [Online]. Available: <https://seprated.github.io/Assessment2/backlog-3only.pdf>. [Accessed 19 January 2017]
- [21] SEPR, "Plan," [Online]. Available: <https://seprated.github.io/assessment4.html>. [Accessed 19 January 2017]
- [22] SEPR, "Backlog," [Online]. Available: <https://seprated.github.io/Assessment3/backlog-4only.pdf>. [Accessed 20 January 2017]