Method Selection and Planning

Method Selection

We have identified the Agile method, specifically the Scrum framework, as the most appropriate development method, since it works well in small teams [1, p.59] and suits the needs of a flexible project environment. Additionally, the Agile method aids our lack of experience in software engineering since it focuses on delivering small releases through short iteration cycles which allows us to adapt how we work as we progress and learn. Whereas with a plan-driven method we would struggle to adapt as we are committed to a fixed plan. Our choice of an Agile method promotes cooperation with the customer [1, p.60]. This benefits the project as it allows the customer to evaluate iterations of the product and to check that the system meets the requirements. Another important aspect to consider is that an Agile method grants us greater flexibility and allows us to adapt to any requirement changes [1, p.60], with less of an impact on the project than if an alternative method was used.

We chose the Scrum framework because part of the team have previous experience using it. This will enable them to support the other team members if there are any issues with using the framework. Additionally, Scrum is a very solid and successful approach that is commonly used in many companies, a survey of 120 companies found that 54% of companies using an Agile method used Scrum [2, p.34]. Scrum is also benefits developers as they can see the progress of the product happening quickly during the sprints; reducing the chance of the team feeling like the product is not making any progress and causing people to lose enthusiasm.

In practice such an approach will be carried out by having frequent team meetings, two to three times a week, during which we can review and monitor the state of the current sprint as well as plan future ones. Meeting in person and constantly evaluating our progress is something that is extremely important since we want to make sure we are able to stay on track and push ourselves to increase the team's effectiveness. Additionally, we intend to regularly meet with the client so that he can validate the project's current iteration and check the system is meeting their requirements. The length of the sprints will vary throughout the project, during Assessment 1 the sprints will be shorter, from three to four days, while in the subsequent assessments sprints will last one or two weeks to better fit the nature of longer and more complex tasks such as implementation and testing.

Tools

To support the project we will be using a range of tools. After analysing the brief we found that our team would require tools for: team management, file sharing, version control, code sharing, communication, planning, UML diagram production and Flowchart production. When looking for these tools some of the key criteria we looked for was: free to use and promoted remote collaboration.

Team Management - Jira

For team management we are using Jira, a management tool aimed at Agile teams. The tool allows us to: create a todo list of tasks; set deadlines for when the task must be completed and assign a team member to complete it. These tasks can then be moved between three different sections depending on their progress: in progress if it is currently being worked on; blocked if it cannot be completed due to a prior task being a prerequisite and complete if the task is done. Finally, Jira can produce statistics on the work team members have completed therefore we can review how everyone is working and provide support to any team members that may be falling behind.

Trello

Trello will be used to keep a track of what tasks are available to complete, who has been assigned to active tasks, and which ones have been completed. It will help to organize both the product and sprint backlog artifacts utilized by the Scrum methodology, and also add accountability to different tasks. Furthermore, Trello ensures this information is readily visible to the whole group, and is also compatible with Slack through a plugin.

Slack

We have chosen to use slack to organize formal meetings and communication, and coordinate the usage of other tools such as Trello and Github. It allows the group to communicate with each other or individually, and has both web based and mobile clients. Slack can also set and remind people of meetings through Google Calendar.

File Sharing - Google Drive

Google Drive was chosen for our file sharing tool as it provides comprehensive collaboration support for working on documents within a team. This means that several team members are able to work on and review documents simultaneously. Also, using this cloud based system reduces the risk R13 [3] of losing data if the files were stored on someone's personal device.

Code Sharing and Version Control - GitHub

For managing code sharing and version control we will use GitHub. GitHub was chosen over similar pieces of software, such as Bitbucket, as we are able to get a free private repository because all the team members are students. Furthermore, GitHub uses a decentralised system: meaning that if there was a failure with the server and a data loss each team member would still have a copy of the project data meaning that it could be restored. This is beneficial over a centralised system as a server failure with this type of system could set the project back significantly if there were no recent backups available.

Communication - Facebook Messenger

Our team required a tool that would allow us to communicate with each other quickly and from anywhere. For this we chose Facebook Messenger as the entire team is familiar with it already and it can be accessed from a range of platforms. Additionally, Messenger has event planning and straw poll tools integrated into it. This allows us to quickly create impromptu meetings that everyone will be notified about and people can mark whether they can attend; in contrast to the regular meetings that are scheduled using Google Calendar where attendance is mandatory. The polling system allows us to make democratic team decision.

Google Calendar

Used for keeping track of meetings organized through Slack via the plugin.

Gmail

Used to coordinate meetings with the customer, and ask questions that are simple enough to be articulated and responded to by email rather than in person .

Google Office Package

Will be used to allow documentation to be written up. Whilst Docs will primarily be used, we will also use Sheets to create a GANTT chart.

GitKraken

GitKraken is a piece of software that provides a user interface for the Git project. This was used by team members to visualise the progress of the development of branches and the overall project, and as a means of version control on their own systems.

Planning - ProjectLibre

To plan how we will spend our time during each assessment we will produce Gantt charts using ProjectLibre. This software was chosen, over an alternative like Microsoft Excel, as it produces the chart automatically after inserting the data. Whereas with Excel we would have to manually produce the chart by colouring cells. The Gantt charts will then be used in tandem with Jira by setting tasks, and their deadlines, based on what needs to be completed for when in accordance with the Gantt chart.

UML Diagram Production - StarUML

To create our UML class diagrams, we decided on using StarUML. It's free, and allowed us to quickly and easily create professional looking diagrams that are clear and easy to understand. Furthermore, it allowed for diagrams to be exported to PNG files which we can easily integrate into our documents.

Flowchart Production - LucidChart

LucidChart is an online tool for producing diagrams. We are going to use it as the software supports collaboration between team members, meaning that multiple team members can work on diagrams at once.

Game Map Design - Procreate

Procreate is one of the best drawing applications for iOS, which supports drawing with a stylus. Since the team graphic designer was already familiar with it, it seemed the best software to use to hand draw a realistic map of the University.

Graphical Interface Design - Photoshop

Before starting to implement the GUI, it was necessary to create wireframes and test our ideas on how to display information on the screen. Photoshop is a powerful image editor which the team graphic designer had much experience with, so it was the best tool to create prototypes of the interface we wanted to develop.

Implementation - IntelliJ IDEA

IntelliJ IDEA is a powerful IDE which offers advanced code refactoring and navigation capabilities, additionally it is available for free for students. Since the majority of the team already had experience with it, it seemed the best integrated development environment to use for the project.

Team Roles

After analysing the project brief and researching software engineering roles [4, 5, 6, 7, 8] we identified the main team roles that we would need to deliver a high quality product: Team Leader, Secretary, Head Developer, Web Developer, Test Leader, Client Interface and Risk Manager. In order to meet non-functional requirements NF2 and NF4 [9] it is necessary to have a Graphic Designer and an Audio Producer. As soon as we started the implementation we recognized the importance of having a team role dedicated to managing version control, in order to keep track of all changes to source code. In addition to everyone's specialist roles all team members will take part in development and testing the software. Finally, it is important to note that the roles are not fixed and we shall adapt the team's structure where necessary to improve the team's performance.

Team Leader and Graphic Designer - Giovanni Ilacqua

As Team Leader, Giovanni takes on the role of Scrum Master and is therefore responsible for: setting team members tasks; managing sprints and resolving team conflicts. Giovanni was chosen for this position as he has previous experience in leading projects and he would like to improve his team leading skills. Additionally, Giovanni is responsible for producing the art assets for the project. These include: designing the logo; producing images for the website and creating the graphics shall be used in the final game.

Secretary and Audio Producer - Owain King

The Secretary is responsible for recording the minutes of each meeting. The details to be recorded are the location, time, attendance, what was discussed and what needs to be completed before the next meeting. Also, Owain was chosen as the Audio Producer for the project as he has prior experience in using Audio Production tools and is musically talented and is therefore the most fitting candidate in the team.

Head Developer - Dominic Taylor

The role of the Head Developer is to coordinate the team during the implementation and enforcing code standards upon the other developers. Furthermore, he should be checking the code is properly documented so that it may be easily updated and extended in the future. Dominic was chosen for this role as he has previous experience in developing games using Java and the LibGDX framework.

Web Developer and Version Control Manager - Ryan Laycock

Due to his previous experience in website creation, Ryan was selected as the Web Developer. He is required to initially setup the website and update it as we complete each Assessment with the new deliverables produced. Ryan also has work experience using git and was already familiar with the base mechanisms of version control, therefore he was nominated Version Control Manager. Among his new responsibilities there are creating a team repository, keeping track of branches and ensuring any conflicts are solved before merging each branch to master.

Test Leader - Jack Radforth

Jack is our Test Leader meaning that he is required to produce a full and comprehensive Test Script which we can use to ensure that the game is running as intended without unexpected bugs and behaviour. This role suits Jack as he has had previous experience producing and running unit tests in Java, using the JUnit testing framework.

Client Interface and Risk Manager - Peter Beckingham

As the Client Interface, Peter must be familiar with the client's needs and requirements to ensure that as development progresses the product is on track to meet them. Furthermore, he must be in frequent contact with the client to ensure that the he is satisfied with the progress of the product. Peter is also the Risk Manager and is therefore responsible for keeping a record of what risks have occurred and manage the respective risk owners during risk mitigation. The roles suit him since he wants to improve his communication skills and he is looking forward to leading a team in the future.

After examining various software engineering roles, we have decided to organise our team as per Scrum [18]. The team is divided into a Scrum Master, Product Owner, Development Team. Team members will be allocated these roles based on their knowledge as well as their expressed desires for the particular role. We are also aiming to minimize formally defined roles, instead we feel it is a better approach to flexibly adapt to the workload as the project progresses.

The Scrum Master will facilitate discussion within meetings by ensuring that the Scrum protocol is adhered to, but we feel it will be more appropriate to define a Team Leader for each section based

on the skillsets of each individual. The Team Leader will take on the role of a more traditional leader and will lead the discussions within meetings by giving direction based on their area of expertise. This means that they will take lead of discussions on content, where the Scrum Master is more concerned with the way in which meetings are conducted. The Team Leader will also be the first port of call whenever a group member is stuck or unable to progress due to a lack of skills.

The other role we will be using, as per Scrum, is the Product Owner. They will be responsible for organising the tasks in the product backlog over the duration of the project, and clarifying any questions with the tasks that the team may have.

Team members will have the option to choose which task they want to complete themselves at the start of each sprint as defined by the Scrum process. This gives team members the optionality to complete tasks they are more suited to, which allow the team to make progress efficiently. We have decided our sprints will be one week long, we will be meeting at least twice a week, once for a sprint planning/review meeting, and once to assess the current progress of the sprint. Naturally, this may increase depending on the workload or conditions of the project, but we feel that formalizing two meetings a week will afford us the best chance of keeping our Sprints on course. Any issues which remain unresolved by the end of a given sprint will be discussed and added to the list in the following sprint, so the team does not fall behind schedule for the tasks that are required to be completed.

With this in mind, our formal roles are:

Scrum Master - Lewis Neal
Project Owner - Lewis Neal
Team Leader - To be determined at the beginning of each assessment and re-evaluated over time

Project Planning

Since we are using the Scrum framework, it is necessary to have a product backlog list [4] outlining all assessment objectives and tasks to be completed throughout the project which, as a team, we assign and manage on Jira.

However Jira alone does not provide a complete overview of the project and does not highlight tasks in respect to a timeline for the entire project. In order to solve this issue we decided to create Gantt charts as they produce a graphical overview of priority and time required for the project as a whole and they also work well in unison with Jira.

As we are using the Scrum framework, it is vital that we keep track of any assessment objectives and task to be completed throughout the project. For this reason we are using Trello which helps us organize the product backlog list used by the Scrum Methodology. We will also be using Gantt charts to provide a graphical view for the project, this will aid team members complete tasks more efficiently.

Explanation of format

The main tasks are in bold and are made of several sub-tasks, which were identified using a Work Breakdown Structure [10, 11, 12]. Each task is associated with a starting date, a finishing date and a priority, which were all established during several team discussions. Priority is denoted by a 1 to 5 scale where 1 is the lowest priority and 5 is the highest. An additional main task, called Revision of Plan, that covers the entire project has also been added to underline the fact that we will continuously evaluate our plans and methods, with the aim of improving the team's effectiveness and making sure to prevent and address any issues that arise. The revision of plan will also be needed because the established lengths of the tasks are just an estimation of the time needed to complete them, which may not respect the actual time that will be required.

Task dependencies are indicated by black arrows, while the critical path is highlighted in red on the Gantt chart. Non-work time is represented by the greyed out area.

Assessments 2's Gantt chart outlines our plans in far more detail than for subsequent assessments. If we produced a highly detailed plan for Assessment 3 and 4 now it would likely need to be rewritten at a later date. This is in accordance with the principles set out in the Agile Manifesto [13].

Assessment 2
The chart [14] covers a time span of approximately 10 weeks, starting on the deadline of Assessment 1 (08/11/2017) and finishing on the deadline of Assessment 2 (22/01/2018).
Since all team members are planning on going on holiday during Christmas we have decided to allocate two weeks, from the 21st of December to the 3rd of January, as non-work time.
Assessment 3
The chart [15] covers a time span of 4 weeks, starting on the deadline of Assessment 2 (22/01/2018) and finishing on the deadline of Assessment 3 (19/02/2018). The plan has been updated to further break down the main tasks required by Assessment 3 and to cover in more detail several sub-tasks.[16]
Assessment 4
The chart [17] covers a time span of 11 weeks, starting on the deadline of Assessment 3 (19/02/2018) and finishing on the deadline of Assessment 4 (2/05 /2018).
The period of eight days, from the 26th of March to the 2nd of April, has been allocated as non-work time, because most team members will be on holiday and to take a break from the project.

References

[1]   I.S. Sommerville. Software Engineering, Ninth Edition. Available:
https://edisciplinas.usp.br/pluginfile.php/2150022/mod_resource/content/1/1429431793.203
Software%20Engineering%20by%20Somerville.pdf   [Accessed: Nov. 3
2017].

[2]   G. Azizyan, M. Magarian and M. Kajko-Matsson, "Survey of Agile Tool Usage and
Needs", Ieeexplore.ieee.org, 2017. [Online]. Available:
http://ieeexplore.ieee.org/document/6005503/. [Accessed: Nov. 7 2017].

[3]   SEPR "Risk Assessment and Mitigation Risky Developments [Online]. Available:
http://www.riskydevelopments.co.uk/documents/Risk1.pdf [Accessed: Nov. 7 2017]

[4]   P. Abrahamsson, O. Salo, J. Ronkainen and J. Warsta, "Agile Software Development
Methods: Review and Analysis", 2002, page: 29 - 37.
https://arxiv.org/pdf/1709.08439.pdf

[5]   U.E. Eriksson. (2012, July, 30th). Is The Test Leader Still Needed In The Agile World?
[Online]. Available:
http://reqtest.com/testing-blog/is-the-test-leader-needed-in-the-agile-world/ [Accessed: Nov.
5 2017]

[6]   M.B. Bria. (2009, Dec, 2nd). Do We Need an "Agile Team lead" Role? [Online].
Available: https://www.infoq.com/news/2009/12/agile-team-lead [Accessed: Nov. 5 2017]

[7]   S.R. Rothwell. (2012, Dec, 7th). Effective Meetings - Keeping a Record [Online].
Available: https://dzone.com/articles/effective-meetings-keeping [Accessed: Nov. 5 2017]

[8]   H.Y. Yrjola. (2016, Sep, 14th). Agile Scrum In Action 4: Reporting and Monitoring
[Online]. Available: https://wunder.io/blog/reporting-and-monitoring-agile-scrum-in-action
[Accessed: Nov. 5 2017]

[9]   SEPR "Requirements" Risky Developments [Online]. Available:
http://riskydevelopments.co.uk/documents/Req1.pdf [Accessed: Nov. 7 2017]

[10]   SEPR "Work Breakdown Structure Assessment 2" Risky Developments [Online].
Available: http://www.riskydevelopments.co.uk/documents/WBSAssessment2.jpg

[Accessed: Nov. 6 2017]

[11]   SEPR "Work Breakdown Structure Assessment 3" Risky Developments [Online].
Available: http://www.riskydevelopments.co.uk/documents/WBSAssessment3.jpg

[Accessed: Nov. 6 2017]

[12]   SEPR "Work Breakdown Structure Assessment 4" Risky Developments [Online].
Available: http://www.riskydevelopments.co.uk/documents/WBSAssessment4.jpg

[Accessed: Nov. 6 2017]

[13]   Agile Manifesto, 2001. [Online]. Available: http://agilemanifesto.org/ [Accessed: Nov. 5 2017]

[14]   SEPR "Gantt Chart Assessment 2" Sid Meier [Online]. Available: http://sidmeiers.me/documents/GANTT2and3.png [Accessed: Nov. 6 2017]

[15]   SEPR "Gantt Chart Assessment 3" Sid Meier [Online]. Available: http://sidmeiers.me/documents/GANTT2and3.png [Accessed: Nov. 6 2017]

[17]   SEPR "Gantt Chart Assessment 4" Risky Developments [Online]. Available: http://sidmeiers.me/documents/GANTT4.png [Accessed: Nov. 6 2017]

[18]   Scrum.org, "Scrum," [Online]. Available: https://www.scrum.org/ [Accessed: Feb. 17 2018].