Method selection and planning

Software engineering methods

In order for development to be successful it is vital to select the best software methodology for a project. Henceforth when deciding on which methodology to follow for our project we considered many factors, including time constraints, our user requirements, the client, the size of our group and project.

To decide this, we looked at several different methodologies including some examples of plan-driven methods which were shown to precisely depend on clear procedures. We found them to have a repeatable and comparable nature which would benefit us, as it encourages project improvements. However, we decided against using this methodology for several reasons. Requirements change over time as businesses and environments in which they operate change rapidly, therefore this methodology is not suited to rapid development. Not only this, but they allow the developers to have less freedom due to the tight constraints of the planned procedures [1].

One example of a plan-driven method that we looked at but rejected was RUP. We rejected this methodology as, after conducting some further research, we found that certain stages in the method were overly complex and had not been well organised. This further confirmed our decision not to choose a plan-driven method [2].

The next methodology we looked at was agile. Unlike plan-driven methods, they prioritise 'individuals and interactions over processes and tools' and 'respond to change over following a plan' [3]. After further learning about this, we decided that an agile method would probably be best suited to our team. This was because of the many benefits an agile method would offer. The idea of using scheduled sprints of 1-4 weeks means features can be developed quickly and frequently. This arises an opportunity to constantly refine and prioritize the overall product backlog [4].

After deciding that we would use an agile methodology, we had to decide which particular method of agile development to follow. The SCRUM methodology was an obvious choice due to its increasing popularity within large organisations. However, we still felt it imperative to further research this method and confirm our decision. SCRUM works by splitting the project into smaller projects and creates a sprint backlog which prioritises certain requirements over others [5]. After the sprint, the team reviews it and changes anything before moving onto another chunk of the backlog. After conducting this research, our decision to use SCRUM was confirmed. For one this methodology is very fast moving iterative so constant feedback is required from the user, which allows changes to be implemented more easily [5].

One of the most important factors in the SCRUM methodology is communication. This is why we have chosen to use Facebook messenger to encourage our team members to express any issues they have, their progression and reviews of other members work. To easily track our project development will be using an application called Asana.

Development and collaboration tools

Version control: GITHub

We decided to use GitHub to store the game in its development phase. This suited us extremely well as it is widely used for open-source projects and allows multiple developers to contribute to the project [6].

UML Tool: ArgoUML

To create our UML diagrams we have chosen to use ArgoUML. We chose this software as it was free and several team members had considerable experience using it.

Communication: Facebook Messenger

In order to keep in contact during the development of the project, we will be using Facebook messenger. We have created a group chat which includes all team members. This chat will be utilised to plan group meetings and discuss any enquiries about certain aspects of the project. We chose this dedicated communication over others as it was already heavily used by all team members.

Documentation: Google Docs

We also needed a platform to store all the other documentation aside from the main project. We considered different internet-based storage platforms such as OneDrive and Google Drive as they enabled everyone to easily access the documents. In the end we agreed on Google Drive. This allows us to view and comment on other members work with ease.

Task organisation: Asana

To distribute tasks within the team we have been using an application called ASANA. This enables us to clearly outline which tasks need to be achieved and allows us to easily manage who will be in charge of implementing each task.

Team organisation

Organisation

When deciding on how best to organise our team, we looked at different types of software that could be of use. After conducting some further research, we decided to use an application called ASANA which can be used to manage team plans, projects and processes [7]. We chose this application as it was easily accessible to all team members, as it is available as both a web and mobile application, and is 'one of the most popular task management apps available' [8]. This allows us to organise who will be in charge of managing and completing each task, while allowing everyone's progress to be tracked by the SCRUM master. Once a task has been completed, it can be ticked off our linked progress list and all team members will be notified of this development. Completed tasks will then be added to a review list and later will then be reviewed by the rest of the team. Normally this will be completed during team members or discussed in our Facebook messenger group chat.

Structure

As we have chosen to follow a SCRUM methodology there is a precise structure that we need to follow. We needed to allocate a team member to act as 'SCRUM master', whose main role is to keep the team focused on its goal [5]. To decide who would be best suited to the role, we discussed each other's main skills and unanimously voted. We are yet to assign long-term roles to other team mates. Currently, we have decided to divide the work equally until we have enough knowledge to assign roles based on each individual's skills. Not only this, but we have created a sprint backlog to act as a 'prioritised wish list' [5] which holds our project requirements. Our sprints will last around a week to ensure that tasks are completed at a constant and fast rate.

Having said this, we believe there are still areas where our project structure and organisation could be improved. One example being the number of team meetings we have planned to organise. Ideally we would want to arrange a one meeting per day, however, after much discussion, we decided this was highly implausible due to other academic commitments.

Project plan

We have chosen to prioritise each task depending on the amount of work, amount of marks and its vitality to the overall project. Each task has been broken down into smaller sections, this will ensure we complete all aspect of the task efficiently and effectively. To organise the development of our project, we have used a Gantt chart, as shown below. This representation clearly shows how long each task should take and the progress that is required before completing further tasks.

A	B	С	D	E	F	G	н	1.00	J	к	L	M
Project section	Priority	Aut Week 7	Aut Week 8	Aut Week 9	Aut Week 10	Vac Week 1	Vac Week 2	Vac Week 3	Vac Week 4	Vac Week 5	Spr Week 1	Spr Week 2
Assessment 2 - Spr Week 3 Mon												
Architecture Report	HIGH											
Concrete architecture showing structure of code												
Justify architecture												
Implementation	HIGH											
Implement code												
Document code					4 4							
Review work			\sim								_	
State features not fully implemented									_ n		∇	
Software Testing Report	HIGH											
Summarise and justify testing methods									\sim			
Brief report on tests												
Provide URLs for testing material												
Update deliverables	MEDIUM											
Requirements												
Methods												
Risk assessment												
Website	LOW											
Update deliverables												
Upload executable of game												
Upload test plan and result												

Critical path for each task is shown in block colour, we will have multiple teams working on different tasks concurrently. Task dependency is show by the arrows, highlighting which areas must be completed before starting other tasks.

Task	
Change report	Summarise approach to change management.
	Justify any changes to documents.
Implementation	Implement documented code.
	Implementation of architecture and requirements.
Website	Update deliverables.
	Upload executable of game.
	Upload test plan and result.
	Upload user manual.

Assessment 3, Spr week 3 - Spr week 7

Assessment 4, Spr week 7 - Sum week 3

Task	
Evaluation and testing report	Explain and justify teams approach.
	Comment on met/unmet requirements.
Implementation	Implement documented code.
	Summarise software modification.
Project review report	Summarise approach to team management.
	Summarise software engineering methods.

References

- [1] Wikiversity, "Plan-Driven Software Development," [Online]. Available: https://en.wikiversity.org/wiki/Plandriven_software_development#RUP_Artefacts_[Artifacts_in_American-English].
- [2] WeblineIndia, "Top 15 software development methodologies with their Advantages and Disadvantages," 9 Feb 2018. [Online]. Available: https://www.weblineindia.com/blog/top-15software-development-methodologies-with-advantages-and-disadvantages/#sdm12.
- [3] E. Carmichael, "Benefits and Pitfalls of using scrum software development methodology," QASymphony, 22 Aug 2018. [Online]. Available: https://www.qasymphony.com/blog/agilemethodology-guide-agile-testing/.
- [4] Segue Technologies, "8 benefits of Agile software development," Segue Technologies, 2016 June 2016. [Online]. Available: https://www.seguetech.com/8-benefits-of-agile-softwaredevelopment/.
- [5] Belatrix Software development blog, "Benefits and Pitfalls of using scrum software development methodology," Belatrix Software development blog, 26 Dec 2013. [Online]. Available: https://www.belatrixsf.com/blog/benefits-pitfalls-of-using-scrum-software-developmentmethodology/.
- [6] Unleashed Technologies, "What is GitHub and how can it benefit yur development team?," Unleashed Technologies, 9 Sept 2015. [Online]. Available: https://www.unleashedtechnologies.com/blog/2014/08/01/what-github-and-how-can-it-benefit-your-developmentteam.
- [7] Online-Asana, "Use Asana to manage your team's work, projects and tasks," Asana, [Online]. Available: https://asana.com/.
- [8] Process Street, "Trello vs Asana: The best project management app in 2017," Process Street, 2 Aug 2018. [Online]. Available: https://www.process.st/trello-vs-asana/.