Module	SEPR
Year	2019/20
Assessment	4
Team	Early Bird
Members	James Little, Ryan Vint, Adam Lynch, Georgina Martin, Kheng Yeoh, Tanay Malde, Cameron Devane-Waters
Deliverable	Project Report

Software Methods and Tools

We picked the Agile Development method using the Kanban framework for its values of “Responding to change over following a plan” [1]. Agile allowed us to create quick iterations of our software to cope with frequent changes whereas other software engineering development methods we researched such as Model-Driven Methods did not. This was due to the Model-Driven software engineering method’s key concepts being about abstraction and interpretation “The software application model is defined on a higher abstraction level and then converted into a working application using automated transformation or interpretations”[2] which were not as important as the flexibility of Agile.

The Agile Development approach allowed us to easily determine the customer’s specifications and what changes needed to be made in our project for every iteration of the software. We put individuals and actions over processes and tools by communicating with our customer early on and frequently throughout the whole process, therefore following the Agile Development Methods key values “Individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation” [1]. We decided to stick with the Agile Development method throughout the entire project as we knew that user requirements would be changed within the process. However, we changed our initial approach of directly communicating with the customer to using the feedback received from submissions to work as communication of what we should improve on for the customer.

We implemented the Kanban framework alongside the Agile Development method which made our workflow follow a card-based system. We chose and decided to remain with this method due to its ease of dealing with change and the organized system of assigning and working on tasks within the project. We also considered using the Scrum framework alternative [3] however decided against it as it was suited towards projects where teams met more frequently. We found this method suitable for the entire duration of the project with the only change of holding Scrum-like meetings whenever changes occured or one of the leaders found progress to be insufficient to get the project back on track.

We used Trello for managing the Kanban board. We used this through the entirety of our project as it was a simple way to keep track and assign all tasks that needed to be done to group members and was very simple to use so there was no need for us to switch.

We used GitHub for managing our code collaboration and versioning and Google Docs for collaboration and backups of our documentation. Both allowed versioning, markup, and edit histories providing a safety net which helped negate quite a few of our assessed risks. Though we did struggle initially with merging branches on GitHub, we learned that was from poor use of GitHub by our team and once we adopted pushing our changes frequently this was quickly resolved. Once that issue was sorted GitHub was extremely quick and easy to use with a plethora of tools within it so we kept using it through the entire assignment.

We initially chose to use both Discord and WhatsApp as tools for communication. Using Discord to hold group meetings, as it allowed for us to all voice chat, share screens and create different channels of communication that could be used by our different subteams. Whatsapp for general communication such as quick team questions or arrangements. However as the project progressed we switched to only Discord as we found that this fulfilled all our communication needs, and the separation of using multiple apps for communicating made all communication more difficult than it needed to be.

Team Management

As we implemented the Agile software development method we chose the Kanban framework for team management as we felt they worked well together. Kanban splits the overall task into cards which take 1 to 3 days to complete. We decided that each of these cards would be created by breaking down the entire project requirements into smaller sections that could be completed in this time frame. As a team, this allowed us an easy method of understanding how we were going to accomplish the overall project specification as it had now been broken down into more manageable chunks. These cards were then placed into different states that we created e.g. To do, Working on, Finished etc. Therefore this was a good team management framework as different cards represented different requirements of the project and these could be viewed and edited at any time by members of our team allowing for good management. Throughout our project we did not decide to stop using the Kanban framework as when new problems arose, requirements were changed or sections had been completed, team members could easily add or move cards to the relevant states and this could be viewed by the whole team therefore giving the whole team an overall view of what work had been done and what else had to be done before the next deadline so we found it very effective.

We also initially decided that alongside the Kanban methodology we would also adapt and implement a small section from the Scrum methodology. This included splitting up the whole team into numerous smaller teams, which therefore allowed for individual leadership within these sub-teams. This would allow for more individual impact as it would allow the smaller voices to be heard and have more influence in their team than they would in an overall large team. It also allowed us more flexibility as Scrum includes a ‘swarm’ team which can swarm to help any teams that may have a blockage in completing their task [4]. We did not change from using our adapted implementation of the Scrum framework as it worked well alongside the kanban framework since the cards could be viewed by the individual teams allowing them to work on sections of the project they feel they are most suitable for. This allowed us to create a higher quality of work and when help was needed, the swarm team offered assistance which we found a useful solution to problems we faced within the sub teams.

Some sections of the scrum framework were not introduced such as daily 15 minute team meetings because we were not meeting this frequently as a team in real life. However we decided that communication throughout the team was important and therefore decided to hold a 1 hour meeting weekly to allow for communication between the teams and to hold meetings on Discord for when we decided we needed to discuss further. We delegated roles within our teams to assist in having these meetings function as efficiently as possible. The roles we decided initially were meeting chair, secretary and librarian to ensure all meetings have agendas, that the results of the meetings are reported properly and that the proper documentation is managed throughout as we believed these 3 aspects were important with team meetings. As the project progressed we did not continue to have meetings every week with clear agendas with documentation since we believed that frequent messaging over a group chat was easier for us and less time consuming, while still allowing clear communication throughout the group.

We initially used Gantt charts to show when certain sections of the project could be completed due to dependencies from other sections. We believed that this would allow us to visualise the importance of certain tasks over others as creating a Gantt chart “helps you ensure that the schedule is workable, that the right people are assigned to each task and that you have workarounds for potential problems before you start”[5]. However as the project progressed we did not find creating Gantt charts as useful because they were time consuming to create and we had other methods in place such as the Kanban framework to show important sections that needed to be completed.

Bibliography

[1]"Manifesto for Agile Software Development", *Agilemanifesto.org*, 2020. [Online]. Available: https://agilemanifesto.org/. [Accessed: 24- Apr- 2020].

[3]"What Is Scrum Methodology?", *Resources.collab.net*, 2020. [Online]. Available: https://resources.collab.net/agile-101/what-is-scrum. [Accessed: 24- Apr- 2020].

[2]""What is Model Driven Development (MDD)", *Mendix.com,* 2020. [Online]. Available: https://www.mendix.com/model-driven-development/. [Accessed: 24- Apr- 2020].

[4]"The Kanban Perspective on Teams", *Scrum.org*, 2020. [Online]. Available: https://www.scrum.org/resources/blog/kanban-perspective-teams. [Accessed: 27- Apr- 2020].

[5] "Gantt Charts Planning and Scheduling Team Projects" *mindtools.com*, 2020. [Online]. Available: https://www.mindtools.com/pages/article/newPPM_03.htm [Accessed: 27- Apr- 2020].