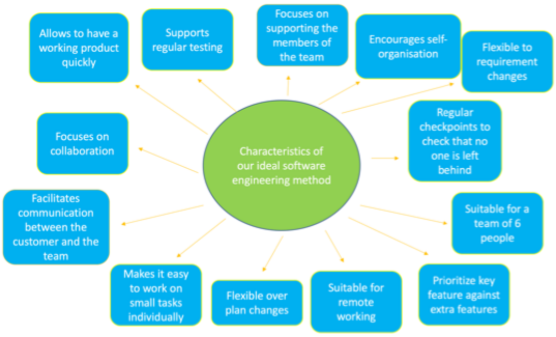
MiKroysoft

Module	SEPR
Year	2019/20
Assessment	1
Team	MiKroysoft
Members	Daniel Crooks, James Rand, Irene Sarigu, Alfie Jennings, Charlotte Clark, Jasper Law
Deliverable	4) Method selection and planning

Section 4a

Early in the project, we had a team brainstorming session where we pointed out the characteristics, we were expecting from our software engineering method. Merging all our ideas we came up with a diagram and we found out that we were pointing to the three pillars, transparency, inspection and adaptation of the Agile's framework, Scrum [1].


Characteristics of our ideal software engineering method
Allows to have a working product quickly
Supports regular testing
Focuses on supporting the members of the team
Encourages self-organisation
Flexible to requirement changes
Focuses on collaboration
Regular checkpoints to check that no one is left behind
Facilitates communication between the customer and the team
Suitable for a team of 6 people
Makes it easy to work on small tasks individually
Flexible over plan changes
Suitable for remote working
Prioritize key feature against extra features

When we investigated more into Agile and Scrum, we found out that they were very suitable for this project. As a team of 6 people with different technical backgrounds and skills, we believe that scrum sprints will help us to plan effectively and ensure that we communicate frequently as this will help us to stay on track with the project. We decided to have two-week sprints as we feel like it's the optimal amount of time to fit in all the agile ceremonies but also is an appropriate amount of time to identify any progress.

One of the most beneficial features of Scru**m** is the daily stand-ups. Whether we decide to do them remotely or in person, every day we will hear what a member has done on the previous day, what they'll do today and if there are any blockers. This is necessary to catch problems immediately.

Our team structure will also have two key roles from a Scrum team, a scrum master and a product owner. We have included other roles, but we feel like those roles were particularly important for the progression of the project and the delivery of the final product.

Due to the nature of this project, requirement changes may occur often during the assessment, therefore, a figure like the scrum master that communicates with the Product Owner and makes sure that "goal, scope and product domain are understood by everyone" is very important [2]. The Scrum master will facilitate the execution of agile ceremonies by making sure that everyone understands the "Scrum way of working" [3] and will also be constantly in communication with all team members to make sure that there are no impediments to the team's progress [2].

The product owner takes care of the product backlog and organises the items within to optimise the work that the development team performs ensuring that the product is delivered as designed and on time [4]. This role is very important to the team because we are developing an application from the very first line of code, there's no code already written which can make the team discussion of what we should prioritise messy and not efficient. This role will ensure that the team always has enough to be working on in any sprint and makes sure that the product backlog is always organised in order of priority. This will give us a good structure as it will make sure that we always complete the basic features (necessary to make the game work) before starting to work on the additional features (adding graphical details to the fire engine).

We decided to divide the tools that we predict we're going to use in three categories: communication, collaboration and planning tools. We put multiple tools in each category for a few reasons. Firstly, we wanted to have options if during the project we found out that a particular tool wasn't as effective as we predicted. This addresses risk RA_NEED from our risk table. Exploring different options now means that we don't need to spend time looking at different tools while we're under pressure to deliver the product, because we already tried several alternatives and we know which option we should use. Secondly, we plan to use different tools for different uses. Especially for the communication category, it's important that we have different tools for each purpose, to make sure team members have organised access to all important information related to the project or our weekly planning (Planning tool graph found at https://mikroysoft.github.io/documentation.html).

The communication tool category contains all the tools we need to communicate with each other. This category includes communication between team members but also communication between the team and the user.

We use Facebook messenger to have conversations that are not strictly work-related such as team-building conversations, last-minute changes in plans or general questions about the assessment. We use this to make sure that all members of the team are doing well with their other modules and that they don't feel too pressured by their work. We use Slack to discuss anything task-related or raise any coding problems that we've encountered. If anyone makes a change in a part of the code where several people are working on, they will put a note on slack and highlight it so everyone knows. Slack is also very useful as we created different channels: one for each assessment, one for technical questions and one to raise any questions related to our workflow, deadlines and meetings with the user. Given that we plan to occasionally work remotely, we also use it to host meetings using the video conference and share screen feature. This feature allows us to communicate effectively when we need to conduct agile ceremonies remotely.

Dividing our communication tools this way ensures that we find a balance between who we are as individuals and the tasks that we need to complete as members of the MiKroySoft team.

For our collaboration tools, we decided to use GitHub and Google Drive.

Most of us had used GitHub prior, so we knew it was a stable and reliable platform. We briefly looked at GitLab but then decided to only use it as an alternative if GitHub goes down. We use GitHub to host our website (through GitHub pages) and to work on the code for our game. This means that the code is open source and that we are able to all work on the same piece of code simultaneously.

We use Google Drive to share documentation, useful resources and anything that's not code. We found this very useful because multiple people can work on the same document simultaneously and the live changes make it perfect for remote collaboration. Another benefit of Google Drive is that you can make suggestions on work files, so a possible change can be reviewed before it has been committed.

For our planning tool, we started by investigating DevOps boards because of its subtask feature. This feature allows you to create an issue and then to link multiple smaller subtasks. During our investigation process, we took some time to set up Azure DevOps. We started using it but we found it hard to set up properly due to permission levels and some of the restrictions present in the free version. We have successfully managed to set up most of our board, but we decided to keep Azure DevOps set up as an alternative and keep investigating for more suitable solutions. We came across the GitHub Project and its Kanban board. These were much easier to use, and they allowed us to have a project for each assessment and a board for each project. The issues can also be carried from one board to another which makes it easy to work with recurring issues.

Section 4b

To make the best use of our competences, we decided to change the usual Scrum structure [3] to a structure that was more suited to our project, experience, skills and personalities. Our team is composed of a scrum master, product owner, head of technology, head of testing, technical writer and architecture lead.

We decided that Irene was the most suited to be the scrum master due to her experience and skills. Irene is currently working in a software company which practises Scrum and she is studying to complete the Professional Scrum Master qualification. Her leadership skills and continuous exposition to the "Scrum way of working" make her suitable to conduct the Agile ceremonies, work with the product owner and "ensuring that the goal, scope and product domain are understood by everyone" [2].

We decided to assign the role of Product Owner to Charlotte because of her previous experience as a head waitress; this involved organising the staff schedule and what task each member of staff needed to complete. During this experience, she learnt what tasks were high-priority, how to prioritise these and how to plan to make sure she could fulfil the low-priority tasks as well as the high-priority ones and so has developed advanced organisational and communication skills. She was also chosen to attend a selection program due to her strong problem solving and leadership skills. These qualities and her experience make her suitable to work on the product backlog and prioritising the tasks.

Our head of technology will be in charge of matters related to technology. He will have the last word on any technology we decide to use, and he'll be responsible for the quality of the product delivered. Daniel was chosen due to his previous experience in leading software teams in his previous university module and the Google Code Hash. He has also worked in a tech company which gives him invaluable technical experience. His technical and leadership skills make him the ideal head of technology for this project.

We chose Jasper as our head of testing due to his previous experience in user testing. During a past degree module, Jasper was tasked with collecting user feedback for in house software from a group of volunteer subjects. Jasper also has experience when gathering feedback from stakeholders as part of a summer software development placement.

The technical writer will be in charge of documentation and code maintaining practises. We decided that Alfie should take this role due to his experience in a tech company where he had to write code in an organised fashion following the commenting and indenting best practices. Alfie also maintained the code for the PROM module last year ensuring that the team created a successful game of pong. As a very organised and disciplined person, Alfie is the most suited for this role.

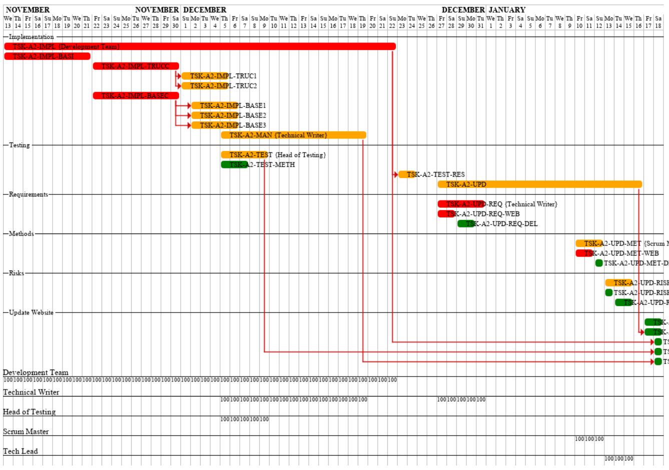
Our architecture lead is in charge of designing and updating the system architecture as the requirements for our project change. He will also be in charge of the structural quality of the code and the quality of our code structure and will work closely with the head of technology. We choose James for this role as he has studied and worked in the architecture field for a long time. He has created architecture diagrams for his computer science A-Level and worked in the Architecture Technology Group at ARM during his internship.

These roles will be fixed but their efficiency will be re-evaluated several times during the assessment to make sure that we have enough time to make role changes without affecting delivery and that the new role-filler has enough time to do some preparation. We will also have a secretary in our team. This will help us when we meet the user as the secretary will be able to take notes and we'll review them during the assessment. This is not a fixed role and we will share it for simplicity.

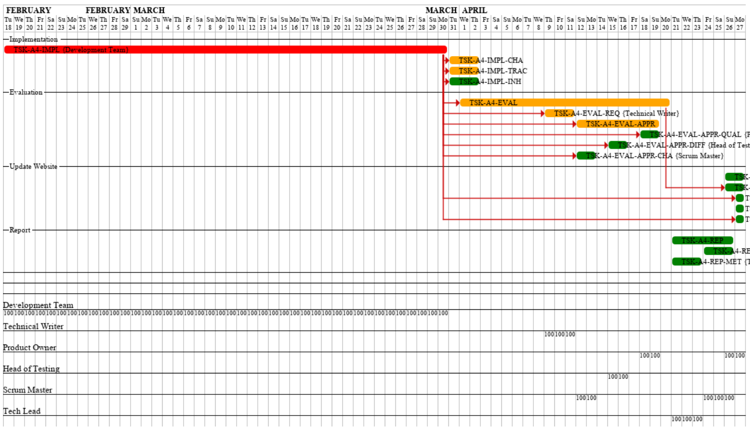
Section 4c

In all assessment plans, implementation was placed before all other tasks. This was due to all other tasks depending on data obtained during implementation. For example, testing.

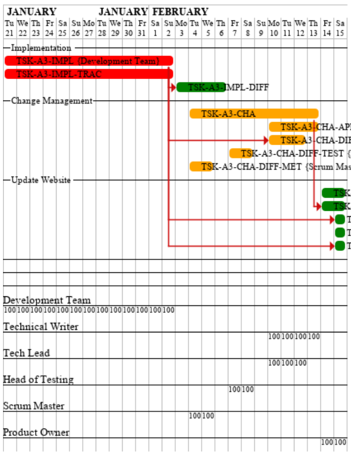
Assessment 2: The base game classes are to be implemented before the classes required by the assessment specification, as this will allow for easier testing of classes. The user manual will be written partially during implementation in the form of basic class names /descriptions, but the bulk of the manual will be written after implementation, to allow more effort to be focused on implementation. Testing can only be carried out once implementation is complete. At this point, we will have the required data to update documentation, reassess and document risks. Finally, the website will be updated with the updated artefacts.



Assessment 3: Traceability documentation is to be written during implementation to ensure completeness. Product change management depends on implementation, for example, in the case of justifying changes made to the product. At this point, the website can be updated with up-to-date deliverables.



Assessment 4: The time required for implementation is not known at this stage, so extra time has been allocated. This will likely be altered once the assessment specification has been released. After the completion of this assessment's implementation, documentation can be completed; changes to the code and GUI to accommodate the requirements change, traceability will be made. The product evaluation also depends on implementation. This is to be completed after the implementation documentation, to allow for effort to be focused as appropriate. The project report is to be completed after the product evaluation, to allow the whole team to contribute to self-evaluation. Finally, the updated deliverables can be pushed to the website. Time assignments may need to be readjusted at a later stage, to accommodate for assessments in other modules and for the assessment four presentation.



Bibliography

[1] K. S. a. J. Sutherland, The Scrum Guide, 2017.

[2] Scrum.org, "What is a scrum master," Scrum.org, 2019. [Online]. Available: https://www.scrum.org/resources/what-is-a-scrum-master. [Accessed 11 November 2019].

[3] M. M. S. Ali, "Scrum Narrative and PSM Exam Guide," 2018.

[4] Scrum.org, "What is a product owner," Scrum.org, 2019. [Online]. Available: https://www.scrum.org/resources/what-is-a-product-owner. [Accessed 11 November 2019].