Method Selection & Planning

Proposed Software Engineering Method

Initially, our team had to choose between the Agile methodology and Waterfall model. However, as the structure of the project implies, the Waterfall model is not the best option for us because it focuses on isolated sequential work – analysis -> design -> code -> test, and it is not particularly good for fast-paced teams. Also, there is a large documentation overhead, which is not suitable for a small team because producing the documentation would be time consuming and is not necessary for non-safety critical applications like ours. Additionally, when using the Waterfall model, it is rather difficult to return to an earlier development stage, such as updating the implementation after conducting system tests, because it can involve modifying a lot of documentation. This is an important consideration because there is a high possibility that the requirements or architecture will change at some point during the project, and so our team decided to use one of the agile methodology methods, called Scrum, as this will be best suited for a small development team, as well as being most capable at adapting to requirement changes.

First, the project is divided into 4 short-period assignments, thus Scrum is perfect for this as it encourages short development cycles, i.e. sprints. Also, we first had to gather all of the requirements for the project from the client and other sources before starting the actual development. In our opinion, the Scrum workflow is ideal for this because within each sprint, a part of the requirements is developed into a shippable part of the product, which can be evaluated by the customer to ensure the product continues to meet their expectations.

Furthermore, a lot of thought was put into choosing a software development methodology, however before coming to a final decision, the structure of the team was decided. We think that as long as the team is managed correctly and work is done on time, there should be no strict structure. Besides, it was already clear that the team members were better suited for working towards all parts of the development, rather than each having an explicit role within the team.
We have decided to tailor the Scrum methodology into a slightly more loose structure. The role of the Scrum master would be roughly replaced by that of the group leader. The group leader would manage work assignments as detailed in the team organisation below, however, unlike a Scrum master, they would not have the greater responsibility for assisting team members with overcoming problems. We decided to split that responsibility amongst the team as it would allow for more dynamic problem solving as we are all of equal experience.
The team itself is self-managing like in Scrum, in most cases without any particular roles with regards to work-loads. However, now that we are beginning to develop our project and code, some of the team (4 members) work mainly on the coding and implementation, with the other members (2 members) focussing more on management and documentation of other areas such as GUI report, requirements management and risk monitoring. This is because when it comes to code development it is easier for those who have already worked with the code to document and test it. If the assessment is more coding heavy then the team will be adjusted accordingly and all members may be working on or alongside code to facilitate documentation.
There will be procedures covered in the team organisation section if this plan goes wrong, for example if deadlines are missed or there are scheduling difficulties.
In order to minimise cases where code is found to be buggy later on in development, unit tests are created after any significant developments in the code.

Development and collaboration tools

To begin with, as clear communication among team members is important and it is not necessarily the case that all work will be done during meetings and practicals, we had to find a suitable way for the team members to work remotely and keep in touch with each other. In addition, although the team is relative small and work-related collisions may be avoided, Scrum suggests that team members are encouraged to work on the same tasks and help each other, thus tools that could ensure high productivity and streamline our workflow are needed. Also, having in mind that it is very common for teams to have a lack of specific end date when using Scrum, we had to find a way to create and manage deadlines in our project. As a result, we chose these development and collaboration tools to include in our workflow:

- **Facebook Messenger** – This is used for group communication outside meetings and practicals.
- **Google Drive** – We use it to store temporary, work-in-progress documents. Google Docs features help us all work on the same document at the same time without any collisions.
- **Github** - We use it to store the source code and final documents. In addition, we take advantage of Github Pages feature and host our website on it.
- **Gantt charts** – Although this type of charts are not mentioned anywhere in the Agile methodology, we still use them to set deadlines for sprints, and, most importantly, track our time during each of them.
- **Email -** For more formal communication, outside of the main team, we will use email. This will be used for contacting the stakeholder with questions or another team if we need assistance with understanding their project.

Team Organisation

As a team we have opted for a fairly flexible plan for organisation that also makes it clear what everyone in the team needs to be doing at any given time.

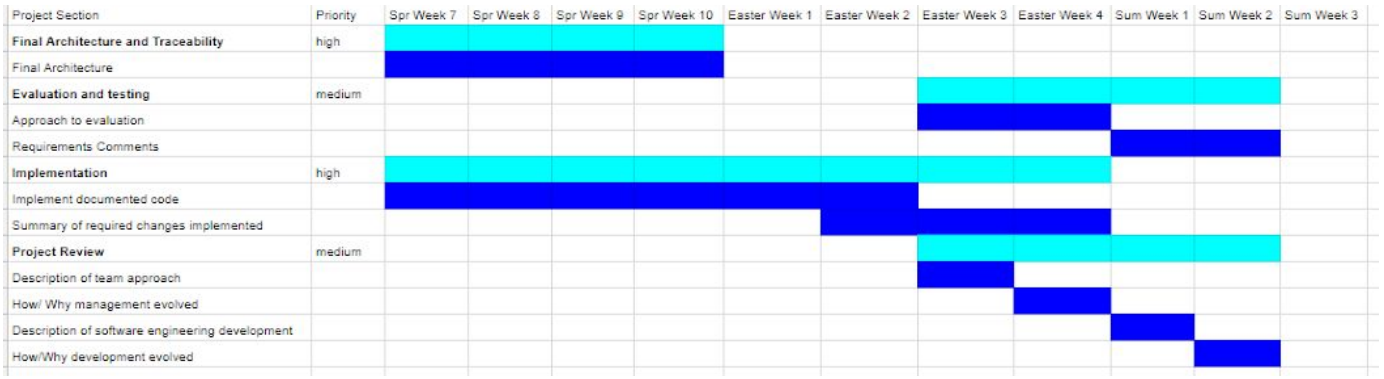
Work is planned for the long term with deadlines for each section through use of a gantt chart. More short term plans are set by allocating work dynamically and is done in sprints. These sprints typically last a week and meetings are set in between so that the whole group can get an overview on what is being covered and offer input if needed.
As a team, during sprints we meet regularly face to face to organise our work deadlines and give updates on what we have been working on. In these meetings we analyse the work that we have done by going through the to-do list as well as discuss any issues that need to be resolved. If anyone is facing any difficulties the team will use these meetings to work together to help overcome them. After or during each meeting a to-do list is updated with the short term details of the sprint and the work it contains, along with information about who is doing it and its deadline.
When starting a new sprint we assign work fairly evenly to each member of the team. If there is a member who feels they are more suited to a given task then they will usually say so and the work will be given to them. This also improves the motivation of the team as they are working on what interests them. The exception to this is when we are developing code - members are still self-organising however the tasks they do will either be code-related or not code-related. The allocated tasks form the backlog of the sprint and a deadline is chosen for them.
When finishing a sprint we review the work and each member will explain the work to the rest of the team so that everyone is up to date with the project. This is also when feedback is given by the team and our progress is presented to the stakeholder if need be. If help is needed from another team to understand any of their documentation, they will be contacted through email or in one of our SEPR practicals.
In the event we begin to fall behind the plan we are following, whether it be due to buggy code, missed deadlines or otherwise, this will be discussed at one of our meetings or in the messenger if it becomes apparent quickly. To resolve it, we will consider how much more work will need to be done on that area in order to bring it up to date with the plan and depending on the severity decide whether someone should assist them, the work should be able to be caught up on in spare time or whether the plan should be adjusted to accommodate the new difficulty.

Our leader ensures that all the work is covered and assigned and encourages the rest of the team to get the work done while participating himself. The leader also manages and prioritises the backlog and verifies that the right work is being done.
The rest of the team does not have any set roles which allows flexibility in work. Each team member is given as much autonomy as possible with their work which ensures a sense of commitment however there is always focus on collaboration - work as a whole is seen as a shared responsibility and the team is usually collocated in its work to help achieve this.
As the team is quite small and there are no great differences in skillsets this seems to be the most natural approach. Meetings are also fairly informal and members of the team are encouraged to put forward their own ideas or ask questions without any pressure.

Work assigned to people within the group is monitored through documentation such as the ownership of risk mitigation and the to-do list but is often done over messenger and meetings as the deadlines are fairly short and without rigid work assignment as long as communication is performed whenever necessary it may be more of a hindrance to document precisely.

Systematic Plan For Rest of Project

Similar to how sprints will have prioritised tasks, overall tasks regarding deliverables for the project will be given priorities too, largely based around the number of marks available for the tasks and any crucial tasks that must be done to ensure the continuation of the project. Each task has been broken down into sections, these sections are based on the questions and the subsections of the questions. We will be making use of Gantt charts, which can be seen below, as they make identifying what task should be done at a particular time very easy because of their clear layout. Each task should ideally be finished when indicted on the Gantt chart, the end of the week for the box highlighted. We have tried to plan around exams to avoid affecting results. We will likely try to finish large tasks, especially implementation as soon as possible to avoid document work piling up towards the end.
The teams approach for Assessment 4 is shown below.



Critical path is shown in cyan, multiple sub teams will be working on different tasks concurrently.