4. Method Selection and

Planning Team 5 | Team Pending

Thom Robinson
Iris Scoffin
Ayman Elsayed
Annice Headman
Izaac Threlfall
Yihong Zhao

A. Software Engineering

Methods and life cycles

- We will use the Agile methodology to develop our software but this does require the team to have good real-time communication and adaption to volatile requirements however, this group has from the outset had the ability and skills to do this effectively but also this method is applicable to this project because it focuses on reducing process overheads and documentation and on incremental software delivery.
- More specifically we will be using scrum as our agile method of choice, this method of development works by using each in person meeting as the start of a sprint, each meeting will be led by a member who will act as the “scrum master” who then will organize everyone else and will also have a rough idea what each member is currently working on. In each meeting we will discuss what everyone has done in the last week and what to work on in the next sprint.
- Also when implementation starts, the fact that pair programming will be introduced will result in expedient code with fewer bugs, pair programming will be used to develop some of the new features that were part of the requirements when we gather for our weekly meetings.

Collaborative tools

- The team is employing multiple tools to complete the project. The project demands collaboration between all the requirements and all team members so, the use of Google Drives allows synchronous, asynchronous, distributed and in-person meetings thus providing the ultimate flexibility and the opportunity for producing the best project possible.
- On top of this, we are using GitHub for the joint implementation of the project and to input our application of the research and requirements elicited in order to create the final product. The Java project will collaboratively and constantly be available to team members. So GitHub was an obvious and effective choice in this regard. This is extremely beneficial as the use of this collaborative tool means we can keep different parts of the project separate but still use the same system meaning they are well organized, effectively managed, easily accessible and navigable by all team members.
- It was decided that GitHub Pages would be used for website creation and maintenance. The focus on end-user performance was ideal for what was required as it simply but effectively displays our project.
- A tool that proved to be really useful was GitHub issues as it helped organize all the tasks that needed to be completed and who was working on them, this made it much easier to stay on track and made teamwork much more efficient.
- Team members mainly used the IDEs they are more comfortable with while using other tools to assist with collaboration like Sourceree and GitHub desktop.
- Discord was also employed to coordinate and help discuss our project virtually and compartmentalize aspects of the project into different channels, which helps to focus on specific parts. Also as different calls can be created smaller groups can collaborate whilst still being in the same server allowing for competent use of the system to advance individual components and then en masse review our progress.

Fitness of tools (comparison to alternatives)

- Google drive was an obvious choice as it provided a slick and intuitive way to store our documentation in a freely accessible and collaborative way. Not only this, it has 15 GB storage capacity which will be more than enough for our project (a clear winner when compared to others e.g. Dropbox). Google Drive syncs across multiple devices and operating systems so team members can always collaborate no matter the device (i.e. members can use iOS tablets and Windows laptops to work together on the same system).
- In GitHub for example, version control is extremely useful. This aspect means tracking changes is easy and identifiable, which is helpful when retrospectively viewing work (meaning asynchronous collaboration doesn't cause redundancy and work can be clearly documented, removing confusion). Also, in certain circumstances, we may need to revert to a previous version, by using this software this can be done extremely easily which ultimately saves a lot of time. Other platforms were available (Bitbucket, TaraVault etc.) but GitHub for many reasons (especially the above) moreover, it is now the most widely used software by developers so, is familiar to the team furthermore, because it is a trusted cloud service provider with security the software element of our project is secure.
- GitHub Pages was compared against others (including Vercel etc.) and the comparison concluded that it would be the most suitable. Some benefits for example are that the website can be changed by team members easily, commits are made with version control, instant version rollback as well as having all other key features in similar tools allowing for effective updation and maintenance of the website.
- Similarly, the team is using IntelliJ to code the game and this software allows for integrated git version control allowing for storing versions if there is ever a need to revert. But it also includes code completion by context analyzing, debugging facilities etc. and has been ranked amongst the best IDEs for java so it would seem to be the optimum tool to use to produce our project.
- One of the overwhelming benefits of Discord is its familiarity. There would be no additional benefits that would help the project from competitor applications (slack etc.) and any other software would likely add superfluous and unnecessary content. The advantages already described (collaboration and designation) are core features that make Discord suitable.

B. Team Organisation

- The team approach to organisation is very methodical and focuses on the acceleration of output without loss of work integrity or quality. Explaining this further gives an in-depth view of the structure: first, the steady start begins, starting attentively was prudent as it lets the team figure out and gather all the information required before commencing. Because of this our team has not produced rushed or erroneous work.
- As the project continues work throughput increases at pace as all members know the tasks they have been assigned and are fully informed. This method allows adequate time for the whole team to review each other's work again adding to quality control and agreement. It is also important to expand on how this method works for the members of the team as well, in this regard members can more easily discuss work, prevent disagreements and be flexible in their work process (as time can be reallocated, see Gantt charts for effectual demonstrations of this).
- There will be allocated 2-hour (term-timed) weekly sessions which will make up a large proportion of review and discussion time but in addition to this, there are also Discord sessions (ranging between 30-90 minutes) to provide a supplementary period to complete and add additional content to the project.
- This way of organisation allows for both in-person and remote contact and along with the style of the two different meetings will define the purpose succinctly i.e. 2-hours will be largely content creation and review and the Discord one will be tying loose ends (which will benefit our team in the long run by keeping us within our time limits and well organised without long periods of unfinished work).
- The structure of our group means that not all members need to be present in meetings so their time can be used more productively and sessions become less cumbersome and more streamlined.
- Having the right balance of people for a specific section is important and the manager/managers of that section can get work done with the non-invasive oversight and outside perspective of at least one other member which engenders a work driven atmosphere but with the security of peer review from at least one other person both during and post writing.
- The hierarchy of the group is flexible and is based on a democratic and fair structure. This way of working may lead to less strong and decisive decision making however, for our team it is and will continue to be effective and due to this way of working a mix of ideas and perspectives are introduced to the project and everyone is as is far as possible pleased with outcomes that are made as a result of a whole group decision. Also, there are managers for different sections and a singular leader might not be able to make specific decisions for each part of the project so, having these managers make important decisions is crucial to the good progress of the project.
- We have created a logbook managed by one member which succinctly describes what has been achieved. This type of organisation helps to keep track of our progression along with the Gantt chart. This means that every week the group can start from the same point and the logbook acts as an external cognition aid to help the team members get on task quicker. All the described organisation above will efficiently and in an equitable way assist our project to run smoothly.

C. Systematic Plan

Knowing if a project can meet objectives on time is essential so having a systematic plan is required. As a group we will draw up a plan based on what should be delivered, when it should be delivered, and who will work on the development of the project deliverables. A plan-based approach requires a stable view of the development processes to accomplish this simple but effective planning method, a combination of written records and visual spreadsheets will be created which will adhere to the Agile methodology but also include the property of visibility. This can and does work for small companies developing software products but like every other professional software development process, Agile development has to be managed so that the best use is made of the time and resources available.[4]

For specific dates and how project modules interact please refer to the weekly updated Gantt chart. These will include key tasks and their start/finish dates.

See Gantt charts for a full breakdown and changes). An example of which is shown below (Second week): The plan for the project requires cascading segments (dependencies) which will be completed by allocated team members but no dead space will be included as the project is constantly being worked on. These dependencies, for example, requirements are a prerequisite for class creation etc. in architecture, are done systematically and logically allowing for the best use of the available time.

As for priority, the current plan is equality, simply to achieve the best quality all parts are required, of course, mark distribution will play a part especially if a component is running behind.

We have set an ambitious and firm timetable but this is to ensure that the project continues to proceed at speed however, this allows for unexpected and unforeseen events because it has flexibility — of course, strict deadlines are still enforced to complete the module on time.

The Gantt chart has been developed to create a more systematic plan. This addition of more characteristics to the key indicates a more thought-through approach and helps to plan our time more effectively.

An advanced Gantt chart is better for our team than a standard chart as the extra information provided removes any ambiguity and streamlines planning. A brief key is shown below to explain how to read these Gannt charts.

Keys	
	Extra Work
	Designated Work
	Undesignated Work
	Work in Progress
	Delayed Work
	Early Start
	Completed Work
*	External Commitments

Weeks:	SPR 5	SPR 6	SPR 7	SPR 8	SPR 9	SPR 10	E 1	E 2	E 3	E4	SUM 1	SUM 2	SUM 3
Dates:	3 - 10	10 - 17	17 - 24	24 - 3	3 - 10	10 - 17	17 - 24	24 - 31	31 - 7	7 - 14	14 - 21	21 - 28	28 - 4
1. Website	Iris												
2. Change Report													
2.1 Summary	Ayman												
2.2 Requirements	Izaac												
2.3 Architecture	Thom												
2.4 Methods and Planning	Ayman	Izaac											
2.5 Risk Assessment	Annice												
3. Implementation													
3.1 Code	Thom Iris	Ayman Annice											
3.2 Documentation	Thom	Iris	Ayman										
4. Testing													
4.1 Summary	Izaac												
4.2 Code	Izaac												
4.3 Report	Izaac												
5. Continuous Integration													
5.1 Summary	Iris												
5.2 Report	Iris												

Key is as below:

- Extra work - Additional work for extra improvement.
- Designated work - Future work timetabled.
- Undesignated work - Timetabled without designation.
- Work in Progress - Work started and will continue.
- Delayed Work - Due to dependency/commitments
- Early Start - Indicates work started before scheduled.
- Completed Work - Finished section.
- External Commitments (*) - Limited time/Restrictions etc. due to external factors.

References

1. Tutorialspoint.com. n.d. *SDLC - Spiral Model*. [online] Available at: <https://www.tutorialspoint.com/sdlc/sdlc_spiral_model.htm#:~:text=The%20spiral%20model%20combines%20the%20idea%20of%20iterative,with%20a%20very%20high%20emphasis%20on%20risk%20analysis.> [Accessed December 2021].

2. Sommerville, Ian. *Software Engineering, EBook, Global Edition*, Pearson Education, Limited, 2016. *ProQuest Ebook Central*, <http://ebookcentral.proquest.com/lib/york-ebooks/detail.action?docID=5185655> [Accessed December 2021].

3. Sommerville, Ian. *Software Engineering, EBook, Global Edition*, Pearson Education, Limited, 2016. *ProQuest Ebook Central*, <http://ebookcentral.proquest.com/lib/york-ebooks/detail.action?docID=5185655> [Accessed December 2021].

4. Sommerville, Ian. *Software Engineering, EBook, Global Edition*, Pearson Education, Limited, 2016. *ProQuest Ebook Central*, <http://ebookcentral.proquest.com/lib/york-ebooks/detail.action?docID=5185655> [Accessed December 2021].

5. Aj141299.github.io. 2022. *Home*. [online] Available at: <https://aj141299.github.io/YorKorsairs/assets/deliverables/Method%20Selection%20and%20Planning%20-%20Logbook.pdf> [Accessed 31 January 2022].

Planning