Method Selection and Planning

Software Engineering Methods and Models

We believe that Agile is the best software engineering model for our team during this project. The flexibility it gives us with changing requirements and the focus on constant interaction with the client will help us create the best product. As a team, we have decided to use scrum as a framework for our project. We decided to use scrum because of how it encourages communication in the team by introducing more frequent meetings. It also helps us deal with requirement changes through lack of committing to a long term design and welcomes any changes to the project backlog. We have divided up the project into "sprints" which last for one week, we chose this time scale over the more common 2 week sprint because of the duration of the various assessments. Before the start of each meeting, we have a scrum stand up meeting which is usually lead by the secretary of the team who asks for an update report from each team member and what they need to get done during this meeting. Then, at the end we discuss what everyone needs to get done during the next sprint. Furthermore, before each scrum, the secretary sends all team members an agenda for the meeting. We believe this will help us to organise our project more efficiently and help keep us on track to meet the deadlines. The list of requirements that we elicited and negotiated from the client is our product backlog and we are taking into consideration that requirements may change.

Development/Collaboration Tools

We have set up a dedicated Google Drive for the project to allow us to share files with each other and ensure everyone has access to the most up-to-date versions of documents in real time. This is useful to us because of the way we divided the tasks amongst ourselves, so if two or more people are working together, they can do so more effectively. It also acts as an automatic backup ensuring work is not lost. Similarly, on Slack we have set up different channels for the separate sub-tasks. This makes it easier for the people working on that sub-task to communicate and share files rather than scrolling through all the messages from the entire team. To collaborate on our code we will use Git hosted on GitHub as this is almost an industry standard. Git allows us to collaborate on code more easily as well as manage past versions of code. GitHub is useful for creating pull requests so that we can review each other's code and also acts as a back up for our data ensuring it is never lost. We will use TravisCI as a continuous integration server to automatically build and test our code as soon as anyone pushes to the GitHub repo. This means when checking a pull request we can already see test coverage and whether tests have passed.

Communication

As a team, the first thing we decided was how we are going to communicate throughout the project with one another regarding meetings, updates and any issues that may occur. We decided to set up a Facebook group chat as we all had Facebook and this is a quick and simple way to keep in contact. In addition, we all made a Slack account which is more business focused than Facebook and specifically designed for team communication. Slack is useful for this project as it allows you to have different communication channels. For example, for assessment one we have created separate channels for all the deliverables i.e. requirements, architecture and method planning. We chose Facebook and Slack rather than email as they allow all the messages to be viewed in one place rather than having to refer back to several, different e-mails. Slack is our main form of communication especially when talking about work to do with the project but the team uses facebook for simple messages like announcing people are moving to a different room for a meeting because of a lab for example.

Team Organisation

Roles

Our main team roles include a leader, secretary and technical lead. Currently our project manager, whose role includes making sure each topic is covered and that we are on schedule, is Robin Stephenson. Secretary, who makes notes of the meetings, sets agendas, checks progress etc.. is Liam Oxley. Technical Lead, whose job is to help other members get up-to-speed with the software and checking the feasibility of our requirements, is Michael Walsh. Note that these team members are not strictly attached to these role descriptions, as we encourage every team to member to help out where they can. These roles can also change hands at any reasonable point in the project at the request of the current roles themselves or other team members. During assessment two, the roles were slightly changed. Team members who were less confident with coding either did pair programming or did the documentation. We allocated these roles depending on the strengths of individual team members as we believe if you are using your strengths you will feel more comfortable and efficient leading to higher quality work. We wanted a structure to the team to ensure that the work is delivered on time and that we are making full use of the team. It also allows individual people to focus on their strength, overall making more efficient use out of our time. The group is happy with this because we agreed our structure would be flexible and that we will make changes if someone is not enjoying or performing in their role, or new roles need to be created. We believe that having some structure however, is extremely important for the project.

Meetings

During the initial phase of the project, we have decided to have at least two 2 hour face-to-face meetings a week. This allows us to reflect back on our progress, share ideas and plan the work that needs to be completed in order to meet the deadline. We decided face to face meetings will be better at the start as we will need to discuss requirements as a team, this will give us the opportunity to get to know each other better. At the beginning of each meeting, we would discuss three questions: 1) What you have accomplished from last meeting? 2) What are you working on this meeting? 3) Do you have any issues to bring up? We then go through the agenda for the meeting and what we hope to accomplish by the end. This helps motivate the team and keep track of deadlines. Then, at the end of each meeting we discuss what each member needs to get done before the next meeting.

Gantt Chart

We decided to use a gantt chart to help split phase two into smaller, more manageable sub-sections and allocate deadlines for each sub-task. Moreover, we have ensured that at least two people are working on a subtask to help reduce risk because if one person falls ill, there will be at least one other person who knows what to do to continue that particular task.

Project Plan

For Assessment 3, we have used a [Gantt chart](#) to provide a more detailed plan of the software engineering tasks.

Assessment	Tasks	Earliest Start Date	Latest Finish Date	Priority*	Dependencies
3	Choose another team's product	25/01/17	29/01/17	1	Assessment 2
	Create a working implementation of the game which meets requirements and concentre architecture	01/02/17	19/02/17	1	Understanding the requirements
	Write up change documents (GUI, Method, testing, risk assessment)	08/02/17	21/02/17	3	Implementation
	Keep user manual up to date	15/02/17	21/02/17	4	Implementation
	Conduct tests and save results	01/02/17	19/02/17	2	Implementation
4	Choose another product	22/02/17	26/02/17	1	N/A
	Implement code	27/02/17	02/05/17	1	Understanding requirements
	Evaluation and Testing Report	20/04/17	02/05/17	2	Implementation
	Architecture and Traceability report	20/03/17	01/05/17	3	Implementation
	Project Review Report	15/03/17	30/04/17	4	Project Completion
	Assessed Presentation	-	-	1	Project Completion

* Higher priority tasks are given a smaller number and are only respective to other tasks within that assessment, not the whole project.