4 – Methods Selection and Panning Sid Meier

Software Engineering Method

Given the wide range of software engineering methods available to us, we were afforded a large amount of flexibility in the approach we took to this problem. After some deliberation however, we concluded that the method that best fit our needs is Scrum[1]. Scrum is a software engineering method for smaller teams that uses an agile approach to project management to break down a project into smaller increments. The team is divided into a Product Owner, Development Team, and Scrum Master, each with their own responsibilities and characteristics.

The Product Owner is responsible for formally defining requirements, and prioritising which ones are completed in which order. They must ensure that the rest of the team clearly understands all tasks and help to determine whether or not a task has been completed. By delegating this responsibility, we ensure that our requirements are accurately implemented into the product as work progresses. On top of this, having one person who is focussed on the requirements means that they can more deeply explore the meanings of each requirements and better clear up unclear areas for the rest of the team.

The Scrum Master is responsible for teaching the team about the Scrum processes and ensuring that both the development team and product owner correctly adhere to their roles throughout the development cycle. Again, we felt that by delegating this responsibility to an individual, the rest of the group would be able to focus more on the content of the project, rather than having the worry about correct scrum protocol. Due to the lightweight nature of scrum, we expect that the amount of additional responsibility attributed to the scrum master will decrease with time, as the rest of the team become more familiar with the process. However at the beginning we feel it is important that the process is adhered to correctly to ensure the most efficient working practices. The scrum master is one key reason that we chose scrum over other agile processes, such as Extreme Programming [2,3,4], which do not have such a role.

The development team is simply responsible for the rest of the development process, and take on all other responsibilities associated with it.

On top of the roles that come with it, scrum also offers us a software engineering method that uses the agile process. This means that we are able to flexibly decide what items should be implemented in what order. As a result, we are able to deal with changes in requirements or unforeseen circumstances without the need to fully re-write a plan, as would be required in non-agile approaches such as the waterfall method. Furthermore, scrum organizes the project into sprints, which allow the team to evaluate progress regularly and make changes to different elements of the project accordingly. Even within these sprints, tasks to be completed are broken down into workable chunks and prioritised as determined by the group. This gives us a structure to measure and make progress effectively, as members of the team can focus on completing tasks rather than figuring out what to do next, as all the tasks will have been determined at the beginning of the sprint.

Finally, scrum organizes contingency planning into the development process, by recycling incomplete items from one sprint back to the top of the product backlog. This ensures that work is completed based on its initial priorities, and a self-contained increment of the project is always ready for release to a reasonable extent.

Development and Collaboration Tools

- Slack
    - We have chosen to use slack to organize formal meetings and communication, and coordinate the usage of other tools such as Trello and Github. It allows the group to communicate with each other or individually, and has both web based and mobile clients. Slack can also set and remind people of meetings through Google Calendar.

- Trello
    - Trello will be used to keep a track of what tasks are available to complete, who has been assigned to active tasks, and which ones have been completed. It will help to organize both the product and sprint backlog artifacts utilized by the Scrum methodology, and also add accountability to different tasks. Furthermore, Trello ensures this information is readily visible to the whole group, and is also compatible with Slack through a plugin.
- Github
    - Github allows us to practice effective version control, keeping track of all versions, past and present, whilst also keeping a record of where and when changes were made between versions. It also shows who made changes to things, and ensures that only members of the group are able to access our code.
- Facebook Messenger
    - Used for less formal coordination and communication. This will typically include organizing meetings on a less formal level once they have already been formally defined through Slack. An example of this would be people letting each other know when they have set off or arrived.
- Google Drive
    - Documentation for our project that is not required to be uploaded to the website will be stored on Google Drive. This ensures that all group members can access it readily, and that it is not able to be lost or damaged.
- Google Calendar
    - Used for keeping track of meetings organized through Slack via the plugin.
- Gmail
    - Used to coordinate meetings with the customer, and ask questions that do not require the extended dialog afforded by a face to face meeting.
- Eclipse
    - Used for the implementation of code within the project. We have chosen to use Eclipse as everybody in the group is familiar with it as a result of first year.
- Google Office Package
    - Will be used to allow documentation to be written up. Whilst Docs will primarily be used, we will also use Sheets to create a GANTT chart.

Team Organisation Methods

For completing the work in section 1 we decided to split the assessment criteria in two, then assign a team of three to each section. We then worked together within these teams to distribute work evenly and ensure that each section of the assessment is completed promptly. Since some sections required group wide discussion we did meet relatively frequently to ensure that each member of the group was able to put across their point of view. Once all of the work was complete, we met again to finalize our documentation prior to the first deadline.

In the early phases of assessment two we aim to work closely as a complete group to ensure that larger overarching decisions in the project are made based on democratic decision. This facilitates a more open and fruitful discussion and should cultivate the best ideas possible for the project. Once our product backlog has been defined as per the scrum process, we will continue to work together as a group during sprint planning meetings, but once a sprint is underway it is the responsibility of individuals to flexibly complete tasks either on their own or in collaboration with other group members. Tasks will be selected by team members from the sprint backlog during sprint planning meetings, but how they are completed will ultimately be up to them. Team members may choose to collaborate on certain larger tasks, though the key point remains that tasks are taken on by the group members themselves, not distributed by a group leader, as defined in the Scrum process.

The Scrum Master will facilitate discussion within meetings by ensuring that the Scrum protocol is adhered to, but we feel it will be more appropriate to define a Team Leader for each section based on the skillsets of each individual. The Team Leader will take on the role of a more traditional leader and will lead the discussions within meetings by giving direction based on their area of expertise. This means that they will take lead of discussions on content, where the Scrum Master is more concerned with the way in which meetings are conducted. The Team Leader will also be the first port of call whenever a group member is stuck or unable to progress due to a lack of skills.

The other role we will be using, as per Scrum, is the Product Owner. They will be responsible for organising the tasks in the product backlog over the duration of the project, and clarifying any questions with the tasks that the team may have.

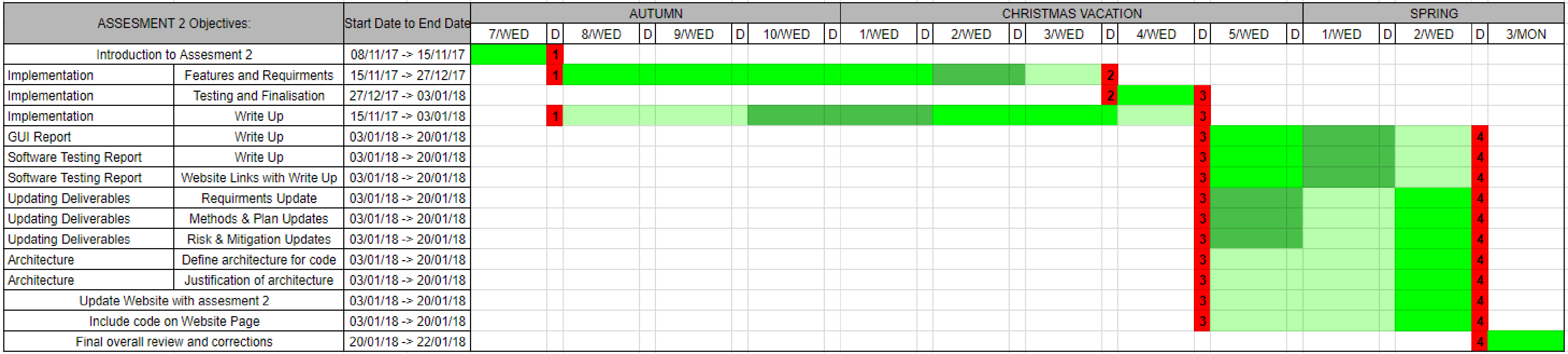
We are aiming to minimize formally defined roles, instead feeling it is a better approach to flexibly adapt to the workload as the project progresses. Whilst consideration was given to Belbin’s team roles [5], we feel the team is too small to limit people to specific roles, as this will bottleneck our ability to complete work more than the alternative of having team members performing of a jack of all trades role. Each team member should be responsible for the completion of the work they have signed up to do in the sprint planning meeting, and although at the end of the sprint all the work will be reviewed and checked over, it ultimately must fall on each group member to choose tasks within the sprint that they feel comfortable with completing, whilst also giving respect to the fact that workloads must remain even. This means that it is expected of group members to make their work available to other group members through the appropriate channels during and after completion of their current task.

Since we have decided our sprints will be one week long, we will be meeting at least twice a week, once for a sprint planning/review meeting, and once to assess the current progress of the sprint. Naturally, this may increase depending on the workload or conditions of the project, but we feel that formalizing two meetings a week will afford us the best chance of keeping our Sprints on course.

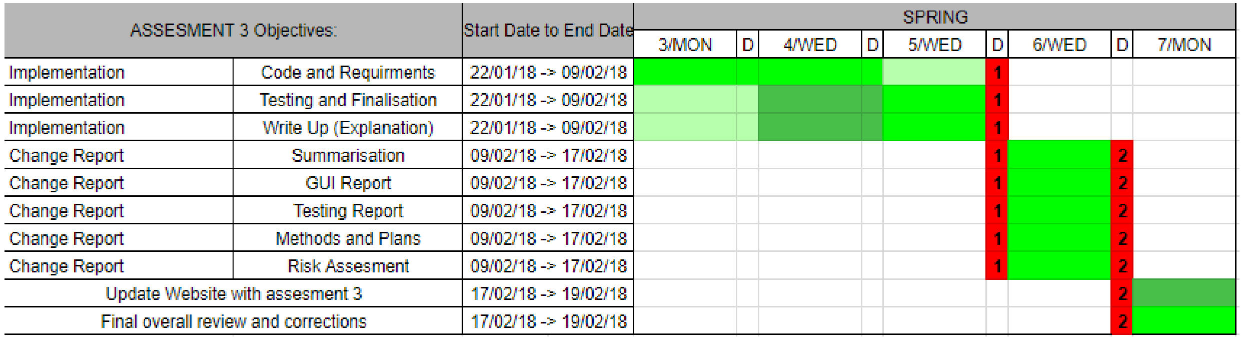
With this in mind, our formal roles are:

- Scrum Master - Lewis Neal
- Project Owner - Daniel Kosky
- Team Leader - To be determined at the beginning of each assessment and re-evaluated over time

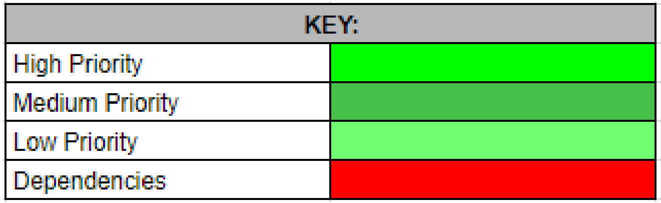
This is our plan including 3 GANTT charts for the rest of the project. Here it outlines each key tasks, more descriptive in assessment 2, for each assessment. Dependencies are marked with red and our numbered to show what needs to be finished before the next tasks can begin. A key can be found to show each priority of tasks as well.



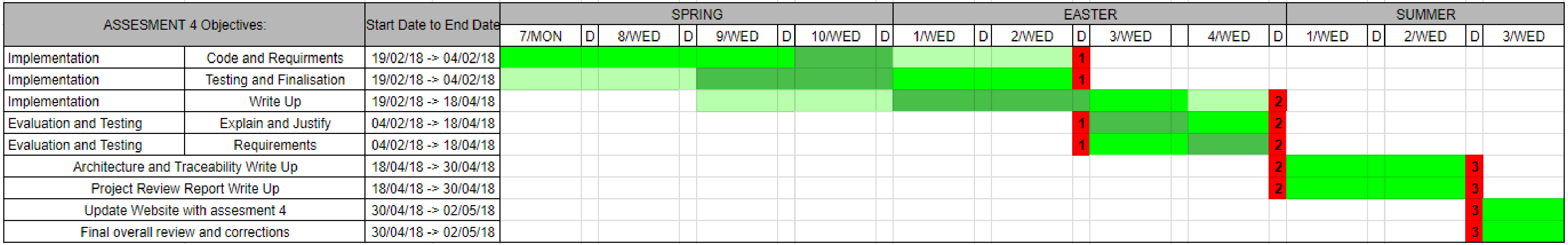
ASSESMENT 2 Objectives:		Start Date to End Date	AUTUMN								CHRISTMAS VACATION										SPRING				
			7/WED	D	8/WED	D	9/WED	D	10/WED	D	1/WED	D	2/WED	D	3/WED	D	4/WED	D	5/WED	D	1/WED	D	2/WED	D	3/MON
Introduction to Assesment 2		08/11/17 -> 15/11/17	🟩	1																					
Implementation	Features and Requirments	15/11/17 -> 27/12/17		1	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	▓	▓	░	2									
Implementation	Testing and Finalisation	27/12/17 -> 03/01/18														2	🟩	3							
Implementation	Write Up	15/11/17 -> 03/01/18		1	░	░	░	░	▓	▓	▓	▓	🟩	🟩	🟩	🟩	░	3							
GUI Report	Write Up	03/01/18 -> 20/01/18																3	🟩	🟩	▓	▓	░	4	
Software Testing Report	Write Up	03/01/18 -> 20/01/18																3	🟩	🟩	▓	▓	░	4	
Software Testing Report	Website Links with Write Up	03/01/18 -> 20/01/18																3	🟩	🟩	▓	▓	░	4	
Updating Deliverables	Requirments Update	03/01/18 -> 20/01/18																3	▓	▓	░	░	🟩	4	
Updating Deliverables	Methods & Plan Updates	03/01/18 -> 20/01/18																3	▓	▓	░	░	🟩	4	
Updating Deliverables	Risk & Mitigation Updates	03/01/18 -> 20/01/18																3	▓	▓	░	░	🟩	4	
Architecture	Define architecture for code	03/01/18 -> 20/01/18																3	░	░	░	░	🟩	4	
Architecture	Justification of architecture	03/01/18 -> 20/01/18																3	░	░	░	░	🟩	4	
Update Website with assesment 2		03/01/18 -> 20/01/18																3	░	░	░	░	🟩	4	
Include code on Website Page		03/01/18 -> 20/01/18																3	░	░	░	░	🟩	4	
Final overall review and corrections		20/01/18 -> 22/01/18																						4	🟩



ASSESMENT 3 Objectives:		Start Date to End Date	SPRING								
			3/MON	D	4/WED	D	5/WED	D	6/WED	D	7/MON
Implementation	Code and Requirments	22/01/18 -> 09/02/18	🟩	🟩	🟩	🟩	░	1			
Implementation	Testing and Finalisation	22/01/18 -> 09/02/18	░	░	▓	▓	🟩	1			
Implementation	Write Up (Explanation)	22/01/18 -> 09/02/18	░	░	▓	▓	🟩	1			
Change Report	Summarisation	09/02/18 -> 17/02/18						1	🟩	2	
Change Report	GUI Report	09/02/18 -> 17/02/18						1	🟩	2	
Change Report	Testing Report	09/02/18 -> 17/02/18						1	🟩	2	
Change Report	Methods and Plans	09/02/18 -> 17/02/18						1	🟩	2	
Change Report	Risk Assesment	09/02/18 -> 17/02/18						1	🟩	2	
Update Website with assesment 3		17/02/18 -> 19/02/18								2	▓
Final overall review and corrections		17/02/18 -> 19/02/18								2	🟩



KEY:	
High Priority	🟩
Medium Priority	▓
Low Priority	░
Dependencies	🟥



ASSESMENT 4 Objectives:		Start Date to End Date	SPRING								EASTER								SUMMER				
			7/MON	D	8/WED	D	9/WED	D	10/WED	D	1/WED	D	2/WED	D	3/WED		4/WED	D	1/WED	D	2/WED	D	3/WED
Implementation	Code and Requirments	19/02/18 -> 04/02/18	🟩	🟩	🟩	🟩	🟩	🟩	▓	▓	░	░	░	1									
Implementation	Testing and Finalisation	19/02/18 -> 04/02/18	░	░	░	░	▓	▓	▓	▓	🟩	🟩	🟩	1									
Implementation	Write Up	19/02/18 -> 18/04/18					░	░	░	░	▓	▓	▓	▓	🟩	🟩	░	2					
Evaluation and Testing	Explain and Justify	04/02/18 -> 18/04/18												1	▓	▓	🟩	2					
Evaluation and Testing	Requirements	04/02/18 -> 18/04/18												1	🟩	🟩	▓	2					
Architecture and Traceability Write Up		18/04/18 -> 30/04/18																2	🟩	🟩	🟩	3	
Project Review Report Write Up		18/04/18 -> 30/04/18																2	🟩	🟩	🟩	3	
Update Website with assesment 4		30/04/18 -> 02/05/18																				3	🟩
Final overall review and corrections		30/04/18 -> 02/05/18																				3	🟩

References

[1] Scrum.org, “Scrum,” [Online]. Available: https://www.scrum.org/. [Accessed 8th November 2017].

[2] D. Wells, “Agile Process,” 2009. [Online]. Available: http://www.agile-process.org/. [Accessed 8th November 2017].

[3] D. Wells, “Extreme Programming,” 8th October 2013. [Online]. Available: http://www.extremeprogramming.org/. [Accessed 8th November 2017].

[4] S. W. Ambler, “Agile Modeling,” [Online]. Available: agilemodeling.com. [Accessed 8th November 2017].

[5] Belbin, “The Nine Belbin Team Roles,” BELBIN Associates., [Online]. Available: http://www.belbin.com/about/belbin-team-roles/. [Accessed 8th November 2017].