CMP3111M Software Engineering Assignment 1

Luke Grant – GRA14589948

Extension Authorisation Code: b9670ed4

Report

Contents

SCRUM Methodology

SCRUM is an agile approach to software engineering for developing products and one of the many project management paradigms. SCRUM is applicable to any project that requires deadlines and end goals with a range of requirements. Within the SCRUM framework, products/projects are completed through a series of iterations called sprints. Sprints are fixed length iterations that normally have a timeframe between two to four weeks.

For the SCRUM framework to be applied to a project, there are a list of SCRUM roles which must be taken to be used for this agile style of development. Everyone within the team is responsible for completing the set tasks that they have been given. For this project we have used the following SCRUM team roles:

- SCRUM Master: The SCRUM master is responsible for guiding the team to meet deadlines and to follow the process of development. The SCRUM master takes a leadership role for the team to ensure that these important tasks are met.
- Product Owner: The product owner is responsible for identifying product features and list them in order of priority. Takes a leadership role in terms of the product itself. A product owner can be a member of the team or in some circumstances the customer.
- Development Team: The development team is a small group of people who are responsible for developing the product for the user. The product owner will ask the development team to create a certain feature and the development team have a time frame to deliver this feature.

The average size for a SCRUM team is from five to nine, but for this software engineering assignment we had a small SCRUM team of three, as we were missing one member from our original group of four. As we were a small team, we decided as a group that all of us would consist of each individual role. We had to adopt all roles and take on the task of completing all the work in the time frame given. In reference to the SCRUM master, we understood that on most occasions one person is chosen to take on this role, but again, as we were a small group we all chose to take on this role.

Other Software Methodologies

Waterfall

Waterfall is a step by step, plan driven process where all the tasks are planned. The process behind this methodology is that the next phase should not start until the previous phase has been completed. The process for this type of software engineering methodology is planned out as followed.

1) Requirements Analysis: Requirements are listed from end users and clients. These requirements are well detailed to ensure they can be followed well.
2) System and Software Design: From the gathered requirements, they are individually split into either a hardware or software specification.
3) Implementation: This stage includes the creation of the software. Developers will work on the program towards its final product.
4) Unit Testing: Ensuring that each individual unit matches the requirements specification.
5) Integration, System testing: This is where the program and units are integrated. They are then tested to see if the requirements have been met, given from the end user.
6) Maintenance: Any errors that come up during the testing phase are corrected here. Further improvement of system units are also developed in this part of the waterfall methodology.

Through analysis of the waterfall method and reflecting on the way in which we have completed this project there are several advantages and disadvantages that the waterfall methodology would have. To start with, one of the advantages of using this method of software development is that it forces a structure to the process and team. It allows a strict design structure which has been planned before the start of the process. This ensures that you list everything that you need to do before you start, benefiting the cost and time implications as you're not planning and changing as you go. By changing the requirements, cost and time implications can lead to these features not being as reliable.

A disadvantage of the waterfall method is that it makes responding to change difficult, this is because you must leap back to the beginning to change/update all processes within waterfall. After the project began its process an update was given to us from the client. The user wanted to request a change, requiring that another movie database was added, such as TMDB. Below is a screenshot of the change request we received.

ACME Software Ltd: Change Request

Posted on: Monday, 5 November 2018 12:57:50 o'clock GMT

We have just received a change request from ACME Software Ltd regarding their Movie Information Client - *"It would be great if you could add other movie database API's, such as TMDB (The Movie Database: https://www.themoviedb.org) to the Movie Information Client. This would mean that users could select a movie database prior to searching. This would ensure coverage and allow an alternative database selection if the search produces less than favourable results."*

Can you incorporate this change request in your sprint planning and product backlog following standard change incorporation and discussions within your team.

Since we had a sudden change of requirements, this would fall as a disadvantage of using waterfall. By following the waterfall method, we may have had to wait until the end to change this requirement, this, therefore, delays the development process. Agile methods are better suited to where software requirement changes happen quickly.

Spiral

Spiral is an incremental model in which the processes completed in a spiral framework rather than a sequence of tasks. This type of model is mainly used for expensive and large projects. The spiral model involves the use of prototypes and testing through the different processes (also known as loops). Each loop is known as a phase of the process. Risks are identified at the start and if anymore are encountered during the phases, the spiral processes, 'loops', again so that these errors are fixed and if needed the requirements updated.

In context to the project in hand that we completed for the client, the spiral method can have some positive and negatives affects as a process towards this project. One advantage of the spiral framework is that of the risk handling process. Throughout the development process, spiral is very useful in terms if correcting errors and taking into consideration the risk factors. During the implementation of our program using SCRUM we ran into several errors developing parts of the GUI, this methodology would have been useful as we would continue to loop through the phases until the problem was resolved. Moreover, this methodology would have the benefit of change requests within the requirements specification. As explained within the waterfall development process, an email was received regarding a specification change. Spiral methodology allows accurate integration of new specification changes within later phases.

On the other hand, there are some disadvantages to using the spiral methodology in retrospect to this project. One of the disadvantages is that the number of phases within the spiral method are unlimited, which can lead to a poor management of time within the development process. A second

disadvantage is that it is much more complex than other software development models. This is a disadvantage because within the time frame given we may have found the spiral model harder to understand than SCRUM model, which not only could've affected our time management but the quality of work due to the lack of knowledge on this model.

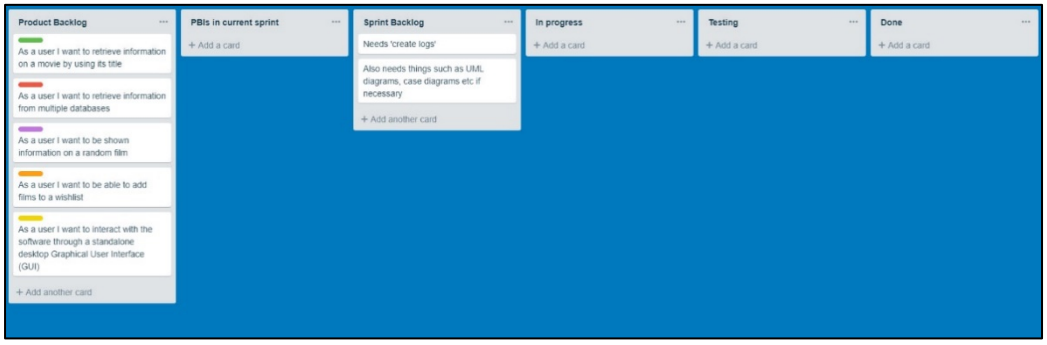
Sprint Logs

For this software engineering assignment, we were assigned the task to create a 'Movie Information Client' for an end user using an API(OMDB). Additionally, we were given a requirements specification which we had to follow to create a full working 'Movie Information Client' to satisfy the client/end user. As a group we chose to break down this task into subtasks, where we could then assign them to different sprints. At first, we gave a time estimation of 10 days per sprint. Below are the three different sprints, with their respected sprint logs, in depth and the tasks fulfilled within each sprint. We decided to use Trello as our virtual SCRUM board for these processes.

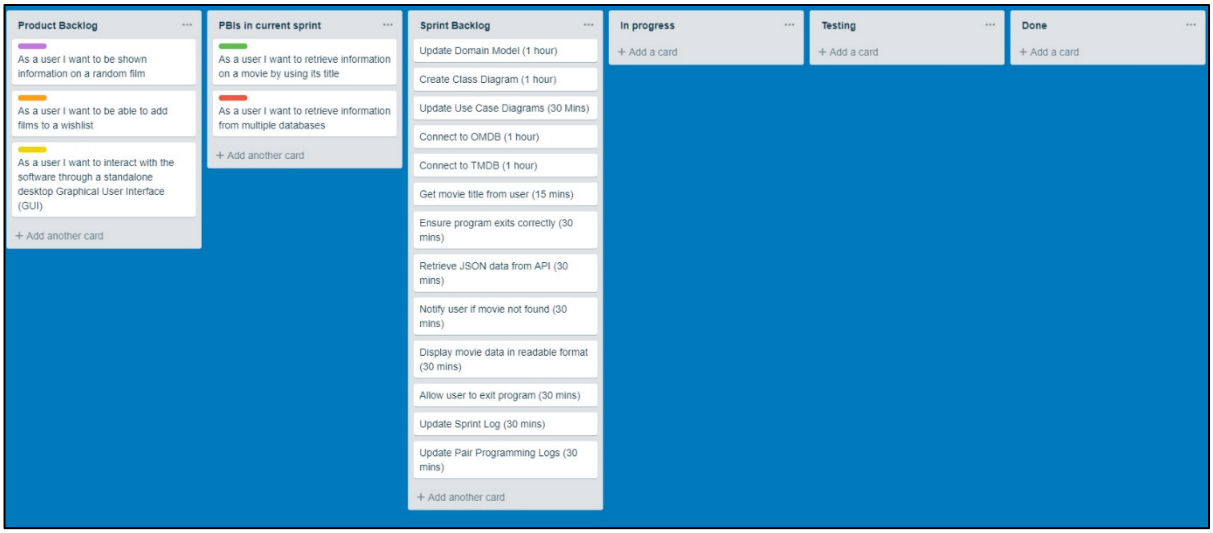
Before we could initially begin our sprints for this development process we had to first take into consideration sprint planning. We started by setting a layout on our Trello SCRUM board, this being the following:

- Product Backlog: Consists of the most important product backlog items, in a stable state, ready to be implemented.
- PBI's in Current Sprint: These are the product back log items in which we would like to complete for the current sprint in hand.
- Sprint Backlog: The sprint backlog consists of the sprint backlog broken down in sub tasks, which focuses on specifics of certain elements of the current PBI. Each task has a time estimation.
- In progress: A task is distributed to a team member; this task is put into this section so that the team can see what tasks are currently being implemented.
- Testing: This is the testing stage of the sprint where we test to see if this current task works. Any errors are fixed within this part of the SCRUM board.
- Done: All completed tasks are placed into the done section of the SCRUM board meaning that these tasks do not need to be revisited as they are complete.

Below is a screenshot if a groomed product backlog shows a set of features of a prioritised list.



Sprint 1



Above is the first sprint in which we established. The Product Backlog Items (PBIs) are the tasks that, we as a group, would like to have completed within the allotted time frame, which for this sprint was 10 days. We decided to include two PBI's in our first sprint as they were familiar in development. This sprint was about retrieving information from an API.

Sprint Log(01 November)

PBIs - Retrieve movie information by title, Retrieve movie information from multiple databases

Tasks	Tasks assignees	Was it implemented	Expected Time	How long	Comments (how this particular sprint went for each task)
Create Domain Model	Cameron	yes	1 hour	2.5 hours	Diagram took longer than expected, we had to fully understand the problem domain
Create Use case diagram	Cameron	yes	30 mins	1 hour	Made use case diagram with basic functions
Create Class Diagram	Cameron	yes	1 hour	2 hours	
Connect to OMDB	James	yes	30 mins	30 mins	Use URL
Connect to TMDB	James	yes	30 mins	30 mins	Getting API key required creating an account with TMDB
Get movie title from user	James	yes	15 mins	15 mins	
Give user a choice of API	James	yes	30 mins	1.5 hours	TMDB returned JSON results differently to OMDB
Notify user if movie not found	James, Luke	yes	30 mins	30 mins	Print statement if JSON is empty
Retrieve JSON data from API	James	yes	30 mins	1 hour	Connect to each API using URL provided
Display movie data in readable format	Everyone	yes	30 mins	1 hour	Format JSON keys into multiple print statments for readability
Allow user to exit program	Everyone	yes	30 mins	15 mins	
Ensure program exits correctly	Everyone	yes	30 mins	15 mins	

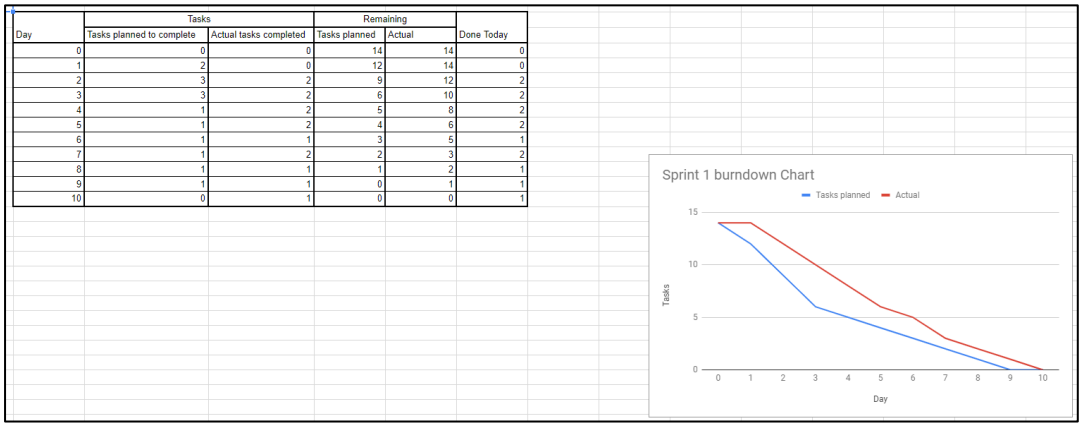
Stand-up meeting (**Sprint Retrospective**): (How the artefact is progressing under SCRUM methodology)

- Coordination - failed to assign tasks appropriately until later in the sprint
- Design phase - created an effective, useful domain model that informed us of the programs basic structure that proved useful in the development phase of the sprint
- Update trello more frequently
- Update sprint logs and diagrams more frequently
- Domain model was well complemented by useful class and use case diagrams
- Time estimation (a problem with SCRUM) - we underestimated the time it would take to complete certain tasks, spend longer in sprint planning phase in coming sprints to address this
- Pair programming worked well with helping members of our group to understand the programming process, and to gain an understanding of the syntax of Python, which some of us were unfamiliar with. This further enabled us as a group to help develop the code, i.e; more efficient coding.
- Need to start using Slack more to facilitate group communications

Here is the screenshot of our sprint log for the first sprint. As you can see and further explained within the burndown chart below, our time estimations were incorrect for some of the tasks within this sprint. Diagrams took, much longer to create/design than expected, this was a drawback for this sprint. Time estimation is one problem of the SCRUM methodology as this process can be only be successful with experienced team members. As this was our first time using this methodology, we fell into this disadvantage of SCRUM. Furthermore, we included a sprint retrospective at the bottom of each sprint to reflect on how the artefact is progressing under the SCRUM methodology. We analysed that, even though we assigned tasks to team members, we failed to display these assigned tasks on the SCRUM board. We further discussed at the end of the sprint that we needed to update the Trello board more frequently, e.g. as soon as we had finished the task. This would lead for us to have a better understanding of which tasks were completed without a form of communication.

Burndown chart (Sprint 1)

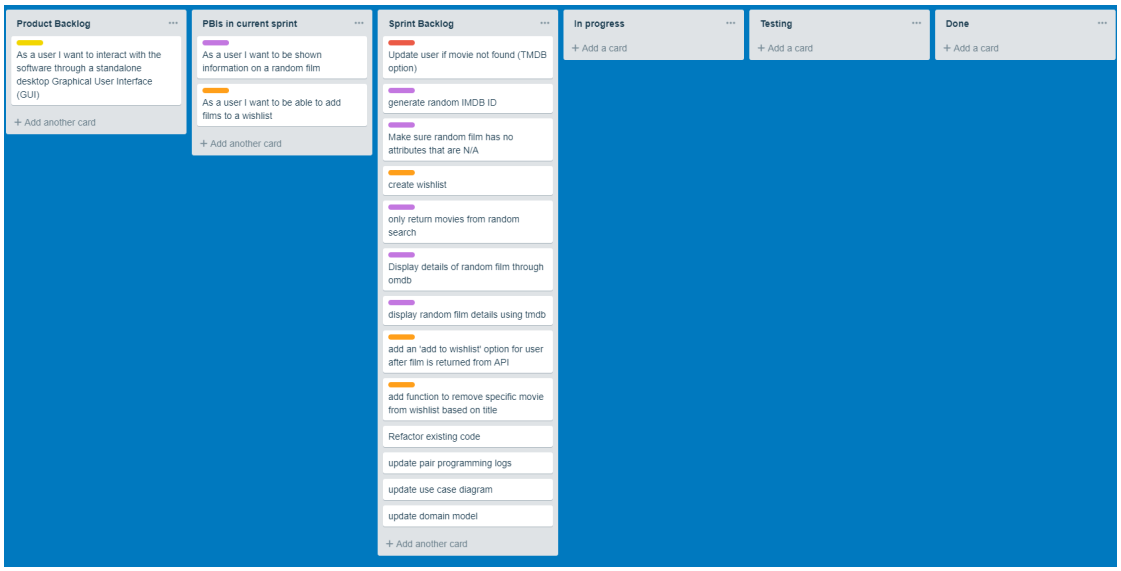
For each sprint we included a burndown chart to see the progress of our sprint. The horizontal (x) axis on the burndown graph represents the time estimation (in days) for this sprint. The vertical axis(y) represents the number of tasks need to be completed within the allotted time frame. Also, on the graph two lines are represented as results, the blue line representing the tasks estimated within the time zone, and the red line showing the actual rate at which we completed the tasks throughout the days of the sprint.



Day	Tasks		Remaining		Done Today
	Tasks planned to complete	Actual tasks completed	Tasks planned	Actual	
0	0	0	14	14	0
1	2	0	12	14	0
2	3	2	9	12	2
3	3	2	6	10	2
4	1	2	5	8	2
5	1	2	4	6	2
6	1	1	3	5	1
7	1	2	2	3	2
8	1	1	1	2	1
9	1	1	0	1	1
10	0	1	0	0	1

As you can see from the graph, throughout the sprint the red line is constantly above the blue. This indicates that throughout the sprint processes we were falling behind schedule. As this was week one of us as a group using the SCRUM methodology is was taking time to start developing more of an understand of this type of methodology. To further conclude, we found that our time estimations for some tasks were too short, meaning that some tasks ran over to the following day of our sprint which meant more of a backlog of tasks to do further down the week. Because of this, sprint 1, ran over the deadline by one day for us to complete the sprint for us to move onto sprint 2.

Sprint 2



Above is a screenshot of or groomed sprint for sprint 2.

Sprint Log 2 (16 November - 30 November)

PBIs - Retrieve random movie, Add movies to a wishlist

Tasks	Tasks assignees	Was it implemented	Expected Time	How long	Comments (how this particular sprint went for each task)
Generate random IMDB ID	Cameron	yes	15 mins	30 mins	Stuck in while loop whilst generating random numbers but fixed shortly after. A quicker method of finding an existing IMDB ID is needed.
Only return movies from random search	Cameron/ James	yes	30 mins	1 hour	Did a check with the chosen API that the ID generated returns content
Display random film details using TMDB	Cameron/ James	yes	1 hour	1 hour	Made different function because keys are different in the JSON from TMDB
Resolve JSON key error conflict in display TMDB function	James	yes	15 mins	15 mins	Put into separate function
Display random film details using OMDB	Cameron/ James	yes	30 mins	30 mins	Made sure to study JSON to understand which keys to use
Create wishlist	Cameron	yes	5 mins	5 mins	Currently an array
Add an 'add to wishlist' option for user after film is returned from API	Everyone	yes	30 mins	1 hour	Pushes film title to array if film is in memory
Notify user when movie has been added to wishlist	Luke	yes	5 mins	5 mins	Print statement
Add function to remove specific movie from wishlist based on title	Everyone	yes	30 mins	1 hour	Using find function and matching characters to array elements Using .remove() did not work initially so we are using del instead
Make wishlist option separate from API	Everyone	yes	1 hour	1 hour	Option given to edit wishlist at the start of the program
Refactor existing code	Everyone	no	2 hours	0	Not done as we decided to tackle this in sprint 3 when we make the GUI
Update pair programming logs	Luke	yes	30 mins	30 mins	Updated pair programming logs for the current sprint
Update domain model	Cameron	yes	15mins	5 mins	Domain model remained the same
Update use case diagrams	Cameron	yes	20 mins	20 mins	Use case includes new operations available to user
Update class diagram	Cameron/ Luke	yes	30 mins	1 hour	Class diagram rework to encompass program features

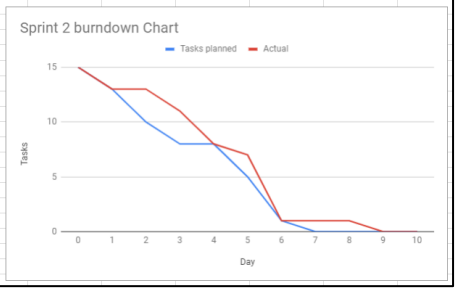
Stand-up meeting (**Sprint Retrospective**): (How the artefact is progressing under SCRUM methodology)

- Successfully integrated additional functionality to the existing artefact (user wishlist and random movie searches) with only several minor bugs to fix during the duration of the sprint
- Use of trello as the project scrum board has become more frequent, with the sprint backlog seeing more regular updates and tasks assigned and completed at more regular intervals
- Team has developed a better understanding of the sprint planning processes, including properly identifying requirements and constraints
- Version control has improved but still requires more communication to ensure that all members of the team work on the correct, up to date branches
- Testing phase has improved with more contributions in spotting and correcting bugs in the program and logging the issues on github
- The sprint has seen better management of the daily scrum activities such as who is working on which task and which tasks remain to be completed, including better decomposition of product requirements with emphasis on high priority tasks
- Sprint 2 has developed the team's understanding of the final end product, as illustrated by improved design/implementation phases
- Longer sprint planning phase has increased coordination within the group

Here is a screenshot of our second sprint log. Sprint 2 consisted of us being able to show information on a random film and to incorporate a wish list, where films can be added. After analysing what could have been improved for sprint one we decided to take this forward into our second sprint, firstly, by improving our time estimations. As shown in the sprint log and the burndown chart, most of our time estimates were correct. Furthermore, we developed a further understanding of the sprint planning process, which was beneficial for us as SCRUM requires a good understanding of its framework in order use this methodology effectively, which we felt, week by week, our knowledge was improving.

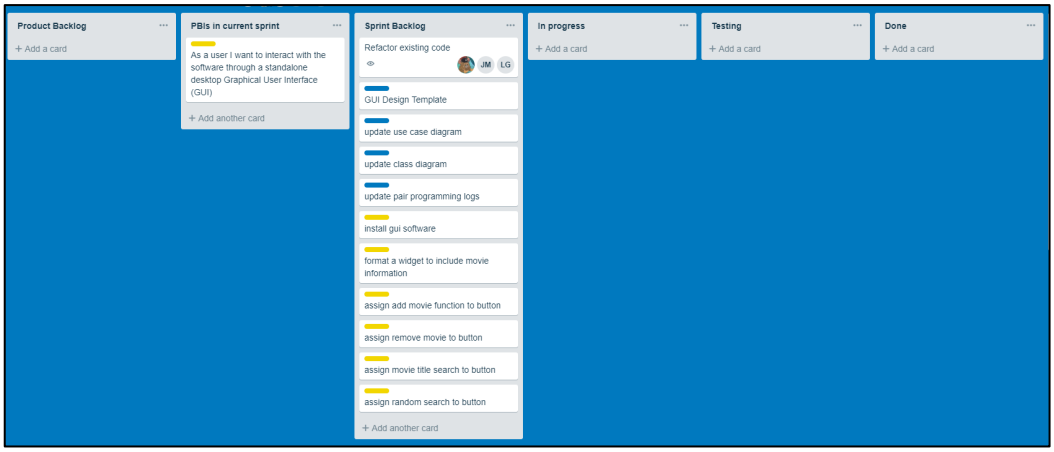
Burndown Chart (Sprint 2)

Day	Tasks		Remaining		
	New tasks planned to complete	Actual tasks completed	Tasks planned	Actual	Done Today
0	0	0	15	15	0
1	2	2	13	13	2
2	3	0	10	13	0
3	2	2	8	11	2
4	0	3	8	8	3
5	3	1	5	7	1
6	4	6	1	1	6
7	1	0	0	1	0
8	0	0	0	1	0
9	0	1	0	0	1
10	0	0	0	0	0



As you can see from the burndown chart above, this burndown chart from sprint 2 looks different to the one in sprint 1. From the analysis of this burndown chat you can see that we still fell behind schedule during some parts of the sprint, but we had made an improvement from sprint 1, which is what we were aiming for.

Sprint 3



Sprint 3 was the last sprint for this project and was the product backlog item that required us to create a GUI. We assigned 12 days for us a SCRUM development team to complete the GUI for this sprint. Above is a screenshot of the SCRUM board for sprint 3, including the PBI in the current sprint, and the tasks that were needed to be completed for this PBI to be complete.

Sprint Log 3 (01 December - 12 December)

PBIs - Ass a user i want to interact with the software through a standalone desktop GUI.

Tasks	Tasks assignees	Was it implemented	Expected Time	How long	Comments (how this particular sprint went for each task)
Create a GUI Design Template	Luke	yes	30 mins	30 mins	Designed a basic GUI with the required specifications of the end user. Did not take long to implement a basic GUI.
Install GUI Software	Cameron	yes	5 mins	5 mins	Did not take very long to install.
Implement Basic Structures of GUI	Cameron/ Luke	yes	1 hour	1 hour	Used pair programming to create the basic structure of the GUI. Pair programming helped us as we were unfamiliar with pyforms.
Format a widget to include movie information	Cameron/ James	no	1 hour	2 hours	Our time estimation was incorrect and therefore we struggled to implement a more complicated way so therefore we had to incorporate an easier solution.
Assign movie title search to button	Cameron(James/ Luke observing)	yes	30 mins	1 hour	This task took longer than expected, we had understand pyforms as all of us were unfamiliar with this framework.
Assign remove movie to button	Cameron	yes	30 mins	10 mins	Easy to implement after our knowledge acquisition from previous tasks.
Assign add movie function to button	Cameron	yes	30 mins	10 mins	Easy to implement after our knowledge acquisition from previous tasks.
Assign random search to button	Cameron (James / Luke observing)	yes	30 mins	10 mins	We implemented this feature but chose to do this using a checkbox rather than a button.
Refactor existing code	James/Cameron	yes	2 hours	3 hours	We adjusted some logic within functions to suit the GUI rather than the procedural programme that we had initially.
Update pair programming logs	Luke	yes	30 mins	30 mins	Pair Programming logs were updated as pair programming sessions happened.

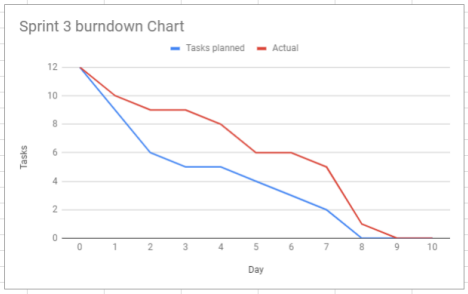
Stand-up meeting (**Sprint Retrospective**): (How the artefact is progressing under SCRUM methodology)

- Utilised programming very well as it helped us even though we were unfamiliar with pyforms.
- Successfully integrated all functionality to the GUI, with some minor bugs.
- We executed a very good sprint planning phase, which increased our productivity.
- Sprint took longer than expected, had to speed up the process during the end of the sprint.
- Github and Trello were frequently used throughout the sprint, with the sprint backlog seeing more regular updates and the github repository been updated more often.
- Pair Programming was a useful process for this sprint as we were able complete the GUI quicker by grouping together as a group to help write/observe the code.

Here is the screenshot of our third and final sprint log. After the completion of this sprint we concluded within the sprint retrospective that we utilised the programming well even though we were unfamiliar with Pyforms. This shows development of our understanding of Python throughout the development as we were able to produce more code with less errors. We also executed a good sprint planning phrase, which increased productivity, this was through analysing after every sprint and constantly improving on the last sprint which we had done.

Burndown Chart (Sprint 3)

	Tasks		Remaining		
Day	Tasks planned to complete	Actual tasks completed	Tasks planned	Actual	Done Today
0	0	0	12	12	0
1	3	2	9	10	2
2	3	1	6	9	1
3	1	0	5	9	0
4	0	1	5	8	1
5	1	2	4	6	2
6	1	0	3	6	0
7	1	1	2	5	1
8	2	4	0	1	4
9	0	1	0	0	1
10	0	0	0	0	0

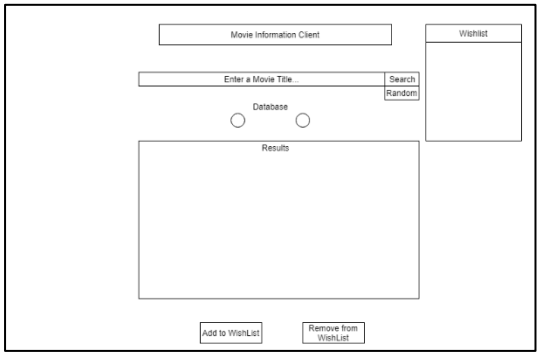


The burndown chart for sprint 3 is similar to the one that represents sprint 1. This was because we fell behind on schedule with the final sprint. This was due to the fact it was our first time using a GUI framework, which we chose to use Pyforms. Due to this we decided that all three of us a group would pair programme so that we could all use our knowledge to put together a GUI. A consequence of this was that it leads to us all focusing on one task at a time, so this slowed our sprint process down. This also meant that towards the end of our sprint we had to rush certain tasks, to ensure that the artefact was finished within the development time zone. This is shown in the table, where we were planned to complete 3 tasks on the last 4 days of the sprint we completed 6 tasks, meaning that we fell behind the planned time for the days of our sprints.

Artefact Design and Implementation

Design:

Below is a screenshot of the proposed design we created for the artefact in which we have created. As a group we decided to create a design template as I gave us an opportunity to save time as we knew how we would have liked the final artefact to look for the end user. To further state we chose



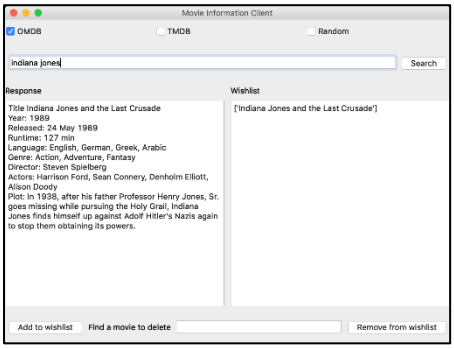
a basic and minimalistic design so that not only was the GUI would be easier for us to implement the final product would be easy for the end user to use. On the design we chose to include two separate buttons so that the user can choose which movie API (OMDB and TMDB) they would like to use. We decided to include a drop-down menu for the Wishlist button so that users can see the movies that they have added. Furthermore, we decided to include a 'remove from Wishlist' button into the design, so that once the users have seen the movie they have added they can then remove that movie afterwards.

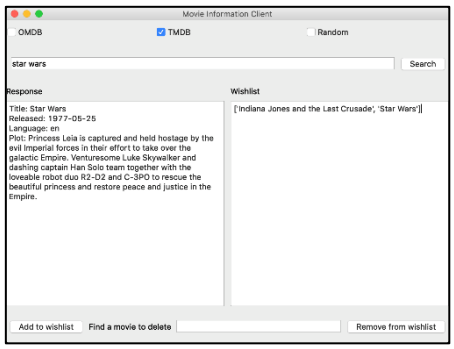
Final Artefact Implementation:

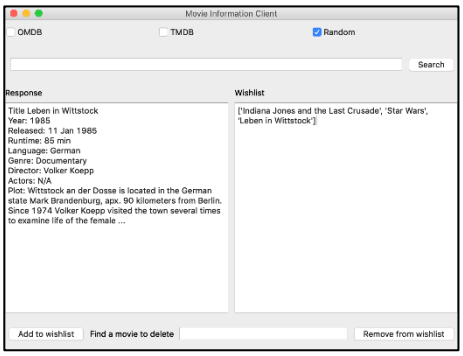
For the Artefact we decided to program the 'Movie Information Client' for ACME Software Ltd in Python. Firstly, we chose Python, because SCRUM works best with object orientated languages. Secondly, we chose Python as a programming language for this program for many different reasons. Firstly, one of them being that Python offers large standard libraries and is useful to build applications with the use of easy to use GUI frameworks. Secondly, we chose Python as all of us had little experience with the programming language, which meant we could all learn and widen our

knowledge of programming within Python. James had the most experience when It came to using Python therefore we had an idea of creating pair programming logs and when we met as a group two to three of us would pair programme.

Below are screenshots of the final artefact, incorporated into a GUI using Pyforms.







The final artefact is near but not perfect in terms of similarity to the design of the movie information client.

Pair Programming

Pair Programming is an innovative practice where programmers work within pairs to develop a software. The way in which this type of practice works is that two or more programmers sit at one computer and help each other program the software. For the instance of our project we chose to use this method of programming as not all of us where as confident when it came to programming in Python.

We chose to create a pair programming log to enter the recorded pair programming sessions. As you can see from the table below we split the log into the following sections; Date(to show which date in which the session took place), Time Spent(this shows the time we spent pair programming for that session), Coder(this person would be the lead programmer for this session), Observer(as there was only 3 of us in the group there would be a maximum of 2 observers each session),Activity(this was to describe what was being coded on that particular session),Number of Lines written, Errors Encountered and how the error was fixed.

Pair Programming Logs - Software Engineering

Date	Time Spent	Coder	Observer	Activity (what's being coded)	Number of lines of code written	Errors encountered (and who spotted them)	How the error was fixed
09/11/2018	1 hour	James McArthur	Luke Grant	Connected to OMDB and TMDB with API keys.	6	Connectivity error, could not connect to the server. (James)	Fixed internet, ensured that the connection was stronger.
09/11/2018	45 mins	James McArthur	Luke Grant	Get movie title from the user and Retrieve JSON data from API	9	N/A	N/A
09/11/2018	10 mins	James McArthur	Luke Grant	Notify the user if the movie can not be found	3	N/A	N/A
19/11/2018	25 mins	Cameron Russell	Luke Grant /James McArthur	Generating a random IMDB title	20	Program stuck in while loop spotted by observer ID not being generated fully before being returned, spotted by observer	Changing use of flags to True, false Indentation changed
22/11/2018	1 hour	Cameron Russell	James McArthur	Fix remove from wishlist function Only return movies from random search	6	Accessing arrays within arrays key errors spotted by Observer Key errors when parsing JSON spotted by	

When partaking in this software engineering project, pair programming was advantages for us as a group. One of the advantages of using pair programming was that as a group (especially myself) we were able to enhance our knowledge of programming in Python. Each session we were able to pick up and learn new syntax and new methods of coding. This then further allowed us to start coding independently after we gain more knowledge of using Python. To continue, a second advantage of using pair programming was that it was easier to spot errors. As we would code in an unfamiliar language four eyes were better than 2. By pair programming we were also able to help each other write more efficient code by suggesting different ways in which we could refactor the code to improve the software structure for the 'Movie Information Client' software.

On the other hand, there were some disadvantages of pair programming. Through analysis of using pair programming we discovered that some of the disadvantages were that it was time consuming. Although there were two/three of us working on the software development on one computer and did increase our learnability and efficiency of coding the software it would not have been quicker than three of us working on the code independently. Overall to conclude, we did find that pair programming was beneficial to our project given the fact that we chose an unfamiliar programming language.

Critical reflection of SCRUM for the development of the Artefact

After using SCRUM as a methodology for this project, it can be analysed as a reliable and good method to develop this project. Firstly, the use of sprints enabled us to break the project down into mini projects which were then further broken down into PBI's then tasks. It enabled us to ensure that all areas were covered and implemented within the project. Sprints were more affective as we were able to use a SCRUM board to manage the sprints. SCRUM boards were affective as we could create different sub topics and assign the tasks, labels, timeframes and assign task to members (using Trello). SCRUM boards allowed the group to keep up to date and notification were sent to one another, when a task card had been moved from one section to the next.

Secondly, within the sprint process of the SCRUM methodology, we found that sprint planning and retrospectives were beneficial for the development of the artefact. This was because we could plan how each sprint would plan out and analyse how each sprint went. In accordance to this, once we had written a sprint retrospective, we were able to improve the way we planned the next sprint, with every sprint becoming more consistent and reliable in terms of deadlines. SCRUM included a lot of feedback throughout its iterative process which was an advantageous principle.

Another advantage of SCRUM we found was that mistakes could easily be rectified, meaning that developing tasks within the sprints became quicker, opposed to using other software developing methods, such as, waterfall. Issues were quickly found and tackled as a group when the tasks were in progress or in testing. SCRUM further allowed sprints to be completed before moving onto the next, meaning that features within those completed sprints were finished and working.

A drawback of using SCRUM methodology was that it took time for us to understand the methodology, in terms of learning the process and getting the processes right. This is in relation to the sprint process, when assigning tasks, time frames and predicting sprint deadlines in terms of the PBI's within the sprint. Some sprints within our development (sprint 1 and 3) ran late by a day or two.

Furthermore, to add to the first disadvantage, we were one team member down, which lead to more tasks being spread over the three of us. As we were unsure of this member of our team turning up, it lead to uncertainty whether to hold back on assigning some tasks to see if this member would attend. After establishing that this member was not present within our group anymore, we overcame this problem by equally assigning tasks to the remainder of the group. This had its consequences though, in relation to the first disadvantage, this lead to time management problems for the first sprint.

Group Analysis

Group Roles

SCRUM methodology requires working efficiently within a team, therefore I have written a small section on Tuckman's model. During our first initial group sessions we decided to engage ourselves

in Bruce Tuckman's, Forming, Storming, Norming and Performing model. We first read and understood what this model meant and how this applied to our group. In brief description the following terms are as described and how they applied to our group:

- Forming: This stage states that during the start of group work most team members are polite and they endorse a lot of effort into getting to know their fellow group members. Members can also feel unsure about what work the team will partake in at this moment of time.

In relation to forming for our group we found that at the start and even throughout we were all positive towards the task in hand. We found it very easy to get along. Although we were unsure about the project at the start, especially with one member down we were confident that we would be able to complete the project within the allotted time frame.

- Storming: In group work every member will have a certain way of working. Within this stage boundaries can be pushed as there maybe conflicts between different members of the group dependent on these work styles. Further conflicts can happen between which roles members have within the group and members may feel uneasy about the workload in which they have been given.

To add to this in accordance with our group, we quickly established our strengths and weaknesses, for example, we all decided to list the things we were good at whether it was from programming to documentation. At this point we did not encounter any conflicts as we quickly identified everyone's strengths and weaknesses at the start. We also ensured that the workload distributed was equal, even though most tasks within the sprints were completed in workshops and meetings.

- Norming: This is the next step from storming in which group members may start to amend relationships where conflicts have broken out. Members of the group start to understand the tasks in hand and most group members working styles. As time has gone on members are becoming more comfortable with one and other and therefore members of the group are more willing to ask for help and there is a stronger relationship to help reach the end goal/product.

As time went on we began to further understand the task in hand set by the end user and the tasks we laid out in our daily SCRUMS. During this stage we felt like the work process was speeding up in terms of completing tasks as we could set task for members in relation to their strengths.

- Performing: This is the stage where most, if not all, group members feel like a committed team, where everyone understands each other's roles and the end goal. Leaders here can feel more relaxed when it comes to designating work as they know the groups strengths and weaknesses.

Finally, this stage represented most of the time in which our group worked as a team. We met deadlines, focused on end goals, and ensured all group members were happy with the task they were set. Help was provided all around the team to ensure every member felt comfortable and confident with their work.

As explained earlier within the SCRUM section of this report, the SCRUM methodology involves different roles within the team. SCRUM roles in which we analysed and used were the following; SCRUM mater, Product Owner and the Development team. As we were a small team, consisting of three members, we decided as a group that all of us would consist of each individual role. Since we

were missing one team member we all had to adopt all roles and take on the task of completing all the work in the time frame given for the end user client.

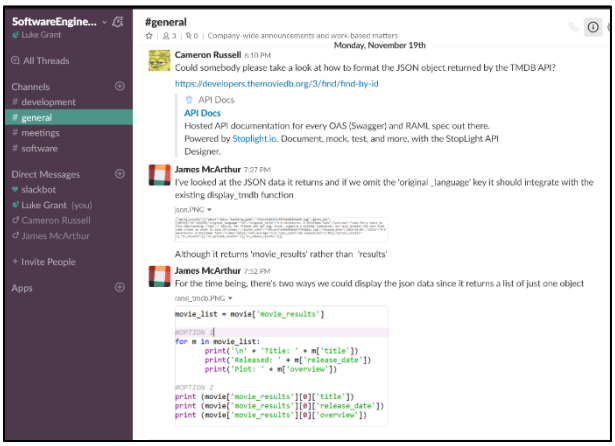
Group Contribution

In terms of group contribution, we all as a group shared the work out equally. At the start we gathered as a group and explained our strengths and weaknesses. I took the role of ensuring that documentation was logged and had a support role within the programming side of the project. From the Sprint logs I was involved in a lot of documentation assisting with the UML diagrams. I established to the group that one of my weaknesses was programming, especially in a new language, such as python. Because of this I ensured that I got involved in a lot of pair programming so that I could pick up on the structure of the code and the syntax, so that I can do smaller programming tasks as the project went on. I found pair programming advantageous, because it allowed me to have a greater impact within programming, especially when we all pulled together to create the GUI.

Tools and Toolsets

Slack (Communication Platform)

Our first point of contact (outside of timetabled workshops) was through Facebook messenger. After establishing that although Facebook is a common way to communicate we were better suited using Slack. Slack enabled us to create separate sub chats which we could label, for example, 'General', 'Software' and 'Meetings'. The benefit of using slack was that we knew where previous messages were stored and allowed us to share documentation easier. The screenshot below shows the use of using Slack as a messaging tool.

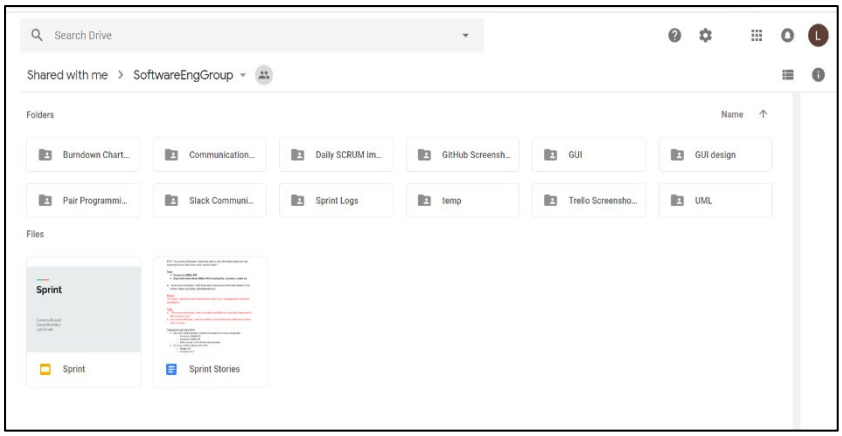


Advantages: As explained above an advantage of using Slack was that we were able to categorise our chats so that when we wanted to revert to old messages we knew where to find them. Furthermore, most of us were familiar with Slack as we used it for our group project in second year, which means our experience of the usability of the communication tool was beneficial to us.

Disadvantages: The first disadvantage we encountered was that often, we kept going back to use Facebook Messenger as a communication platform. As a group we did use Slack when using our PC's to complete our work but when it came to general messaging we did use Facebook. If we were to do the project again we would ensure that we had Slack downloaded on our phones as well as the PC. In conclusion to this, we were not affected by this drawback as there were no communication issues throughout the project.

Google Drive

Google Drive was the obvious choice to facilitate concurrent group work, which provided us a shared mutual working space for our diagrams and logs. We were able to create several folders to categorise our group work to ensure that as a group we were professionally organised.

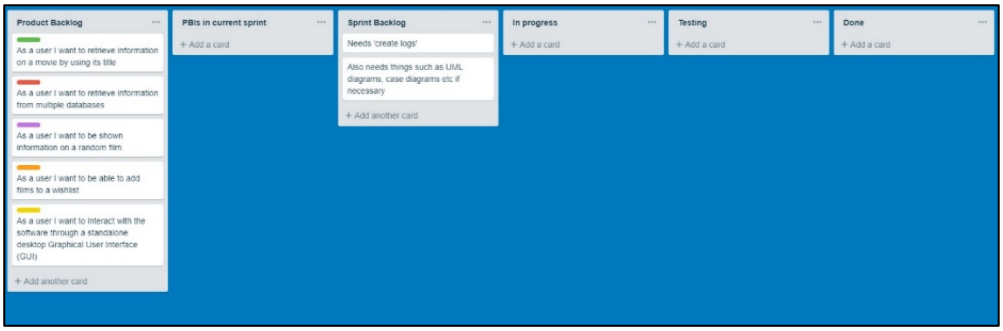


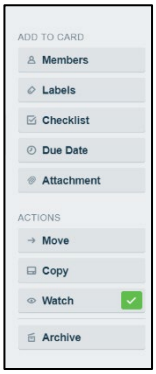
Advantages: The primary advantage of using google drive was to take advantage of a shared mutual working space which saves and keeps work up to date as you work on any of the documents, as it is a free back up data storage. Google Drive also allowed us to store a various range of files from JPG/PNG files to DOC/PDF files. Furthermore, by using google drive we were able to include some of googles workspace features such as 'draw.io' and 'Google drawings'. This was further an advantage for our group as we have all used 'draw.io' before, so by using this tool alongside google drive benefited us as a group.

Disadvantages: When we used Google Drive as a group we had ran into no issues with the platform itself, this was purely because we were all very familiar with the Google Drive platform.

Trello

As a group we chose to use Trello as it offered us a virtual SCRUM board, as this was better than using a physical space and our work could be saved. Secondly Trello allowed us to assign task to individual group members, as well as set time deadlines on each task too. Additional features included drag and drop of the cards, the creation of sections on the SCRUM board and colour coded cards. Below is a screenshot of our first initial product backlog using Trello and a screen shot of some of the features which can be used within Trello that we used as a group.





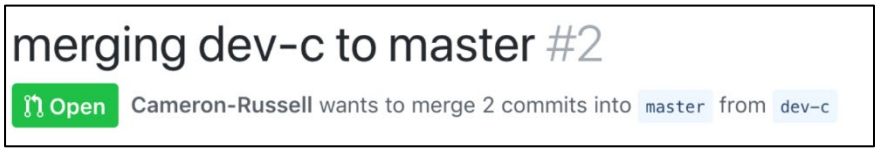
Advantages: Advantages of using Trello were not only was it an easy to use virtual SCRUM board but we were sent notifications every time the board was updated. We were then able to see when they were updated and who by. This allowed us to keep up to date on what was happening when we were individually working on the SCRUM board. Moreover, another advantage was that we were able to keep track of our daily sprints and ensure we knew what tasks we had completed and what tasks we had left to complete.

Disadvantages: The only disadvantage we had as a group using Trello was using the burndown chart extension to the SCRUM board. We found it hard and time consuming to set this up, so we decided to use google sheets instead for the burndown chart.

GitHub:

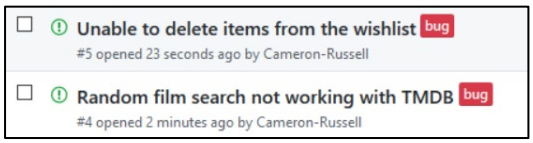
For this development process we decided to use GitHub as it is an open source development site. GitHub is also the world's leading software development platform. The reason for choosing GitHub as a software development platform was because Git is a version control system, meaning that many changes can be made to the code then new versions can be uploaded. This platform worked well with the agile development method (SCRUM) as we needed to keep updating versions of the code throughout our sprints. All updates to the code are kept within a repository, which all team members had access to. Moreover, version control is a great aspect of GitHub as developers can keep up to date and manage the changes that have been made. This allows for developers to see the process of development over a time frame. Branches within GitHub are ways to manage the workflow of the project. The master branch is a default branch where the project is deployable. For this project we created branches called GUI, Classes and dev-c. Branching was beneficial for this development process as not everyone was familiar with GitHub. The reason why we also used GitHub was because as a developer we were able to safely make changes to the code without affecting the rest of the project. To further add, as we were using python, an unfamiliar language to us as a group, we could ensure that wouldn't merge the code back into the master branch until the code had no errors and worked as it was supposed to.

Pull Requests



Above is a screenshot of a pull request Cameron made after a pair programming session involving all of us. Pull requests are a way of notifying other users within repository about change you have made to the code within GitHub. This was a further advantage of using GitHub, meaning that code can be peer reviewed before being pushed through to the master branch.

Bug/Issue Reports



Bug reports were essential for us within GitHub as we were able to monetise the bugs and issues that we were receiving in our code. From this we were able to investigate what the issue was and then go back and fix it. As SCRUM is an incremental methodology, we were able to ensure that before we moved onto the next sprint, we had to ensure that this sprint was finished. The testing phase of our sprints were advantages, as well as the bug/issue reports, because we were able to spot the bugs and ensure that the tasks were put and completed in testing before moving them into the done section of our SCRUM board.

Further Advantages: Branches, as referred to earlier in this section, were beneficial, especially to me, because I was able to work on certain parts of the project without affecting the master branch. I was able to work on a separate branch, then when completed, I would then push the request to the master branch. This resulted in unnecessary errors within GitHub.

Another GitHub advantage as an open source development platform is that it has good documentation. Documentation includes guides to help understand and use GitHub and help section as well if users become stuck with using its platform. To further add, GitHub also have lots of examples on certain projects on how to implement certain features.

Further Disadvantages: One disadvantage of using GitHub was that I was unfamiliar with this software development platform, this meant that I had to learn how to use GitHub well. I understood that when using GitHub, I needed to learn the basics when it came to branches, pull requests and version control. As someone who has never used GitHub before this took time to understand and develop a knowledge on.

References

Acosta, R. and Rodriguez, A. (2018). *Benefits and Pitfalls of using Scrum Software Development Methodology*. [online] Belatrix Software Development Blog. Available at: https://www.belatrixsf.com/blog/benefits-pitfalls-of-using-scrum-software-development-methodology/ [Accessed 15 Dec. 2018].

Agile Business Consortium. (2018). *What is DSDM*. [online] Available at: https://www.agilebusiness.org/what-is-dsdm [Accessed 11 Dec. 2018].

BROWN, K. (2018). *What Is GitHub, and What Is It Used For?*. [online] How-To Geek. Available at: https://www.howtogeek.com/180167/htg-explains-what-is-github-and-what-do-geeks-use-it-for/ [Accessed 12 Dec. 2018].

Finelli, M. (2018). *What Is GitHub and How Can It Benefit Your Development Team?*. [online] Unleashed Technologies. Available at: https://www.unleashed-technologies.com/blog/2014/08/01/what-github-and-how-can-it-benefit-your-development-team [Accessed 15 Dec. 2018].

KUMAR PAL, S. (2018). *Software Engineering | Spiral Model - GeeksforGeeks*. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/software-engineering-spiral-model/ [Accessed 11 Dec. 2018].

Mindtools.com. (2018). *Forming, Storming, Norming, and PerformingUnderstanding the Stages of Team Formation*. [online] Available at: https://www.mindtools.com/pages/article/newLDR_86.htm?fbclid=IwAR3b0vfiJRBVGixc0ym-AbeTk9rdmVCTlUOVKGSD9sWLpL369lBhds9dOwY [Accessed 10 Dec. 2018].

Powell-Morse, A. (2018). *Dynamic Systems Development Method: How it Led to Agile Project Management*. [online] Airbrake Blog. Available at: https://airbrake.io/blog/software-development/dynamic-systems-development-method-agile [Accessed 11 Dec. 2018].

Powell-Morse, A. (2018). *Waterfall Model: What Is It and When Should You Use It?*. [online] Airbrake Blog. Available at: https://airbrake.io/blog/sdlc/waterfall-model [Accessed 11 Dec. 2018].