

SOFTWARE ENGINEERING

CMP3111M



Table of Contents

Introduction	4
Supporting Documents:	4
SCRUM Log Document	4
Product Backlog:	4
Sprint Planning:	4
Sprint Backlog:	4
Daily Scrum Meeting; x4 days a week:	4
Sprint Review:	5
Sprint Retrospective:	5
XP Processes	5
User Stories:	5
Paired Programming:	5
Test Driven Development:	5
Velocity:	6
Estimation:	6
CLASS & RELATIONSHIP DIAGRAMS AND DFD'S	6
Testing, UML Diagrams and Paired Programming from Sprint 4:	7
Class Diagram:	7
Observer Design Pattern:	7
Singleton Design Pattern:	7
Domain Model:	9
Test Driven Development:	10
Dashboard TDD:	10
Randomize Method:	10
Paired Programming Logs:	11
Critical Reflection on the Development of the ACME Client using SCRUM	12
Implementing a SCRUM and its Processes	12
Scrum:	12
Product Backlog:	12
Sprint Planning:	12
Sprint Backlog:	13
Daily Scrum Meeting; x4 days a week:	13

Sprint Review:	13
Sprint Retrospective:.....	13
XP Processes:.....	14
User Stories:	14
Paired Programming:	14
Test Driven Development:	14
Comparison of SCRUM vs Other None Agile and Agile Methodologies	15
Agile Processes:.....	15
SCRUM	15
Dynamic System Development Method (DSDM).....	15
Waterfall	16
Spiral	16
Evaluation of Software Tools	18
GitHub & Git.....	18
What is GitHub & Git?.....	18
How it's use in open source software development?	19
Evidence of Use.....	20
Slack	21
What is Slack?	21
Evidence of Use.....	21
Scrum Desk:	22
Evidence of Use.....	22
Trello:	23
Evidence of Use.....	23
Daily Scrum Meeting: 1/4 (week-1-tuesday-30th-october-2018):	23
Progression Chart: 1/4 (Week 1: Tuesday 30th October 2018).....	23
PyCharm:.....	24
What is PyCharm?	24
How it's been used throughout the development process?	24
Evidence of Use.....	25
Pycharm Local History.....	25
Pycharm VCS using Github	25
References	26

Introduction

Supporting Documents:

SCRUM Log Document

This is a layout explanation for the 'SCRUM Log Document' which has been submitted in supporting documents. It provides a structure of the document, what you'll find within it and explanations of the headings.

Product Backlog:

"Product Owner will register all the initial requirements to be done by the Team (known as the Product Backlog)" (Permana & BALI, 2015)

A product backlog is a table which details all the necessary tasks that need completing in the project. I have followed the DEEP method which has a top level of highly specified user stories and at the bottom general user stories. The top level of the backlog will be the highest priority and will also include story points and a business value. The story point reflects on the user stories complexity, risk, and effort needed to complete. A business value represents the value to the business that the user story has.

Sprint Planning:

Sprint Backlog:

"On the Sprint Planning part, the team members will determine how many hours each member will spend to carry out work on each feature." (Permana & BALI, 2015)

The sprint backlog is a table which is completed during the sprint planning meeting. The planning is the process of taking the product owners ideas of user stories and splitting them up into smaller more completable tasks which can be allocated to different team members based on skill. The table included the product owner's user story, followed by the smaller tasks, then the time allocated to the task and an overall time need to complete the user story.

Daily Scrum Meeting; x4 days a week:

"As a practice to monitor the performance, there should be a meeting every day for reporting what has each member of the team done. At this stage of the Daily Stand Up Meeting not all the team members must be present, but rather the members involved specifically on the features that are being developed (Team)" (Permana & BALI, 2015)

Days in the week which SCRUM meetings are held were Tuesday; Wednesday, Thursday and Friday. These meeting are a short 15 minute stand up meeting where Q&A table is filled out, the progression chart updated and burn chart made. The Q&A asks what has been done in between the last and current meeting, which team member is planning to what tasks next and obstacles from the last day which have prevented progress. The progression chart shows what have been completed and what is in processing and a burn down graph shows the optimal line for working 10 hours a day for 4 days along with how many pieces of work have been completed.

Sprint Review:

A sprint Review is used at the “end of the Sprint to inspect the Increment and adapt the Product Backlog if needed.” (Scrum.org, n.d.). This review is laid out in table representing which of the individual tasks have been completed across the sprint. This review involved “the Scrum Team and stakeholders collaborate about what was done in the Sprint. Based on that and any changes to the Product Backlog during the Sprint, attendees collaborate on the next things that could be done to optimize value” (Scrum.org, n.d.).

Sprint Retrospective:

“The Scrum Team discusses which task is encountering a problem, which task can be continued and that cannot be continued for the next sprint.” (Permana & BALI, 2015)

The retrospective is a table which asks all team members, what went well in the sprint? What could be improved? And what will we commit to improve in the next sprint?

XP Processes

The XP processes that have been included in the SCRUM are: user stories, paired programming and test-driven development.

User Stories:

“Story cards are written by the customer in XP to articulate their business needs.” (Patel & Ramachandran, 2009)

The user stories are an important part of this project, providing the team with knowledge on what the customer wants. These have been used through the development of this project to be used by the team to understand what the end users have ideas about.

Paired Programming:

“More usually, a pair will work together for the duration of a single task that might most often take a day or two to develop. While some projects empower the programmers themselves with responsibility for deciding when to change pair, others enforce rotating pairs on a regular basis. “ (Bryant, et al., 2006)

This XP process use 2 members of a team, allocated to completing one area of the programming. Throughout my scrum a virtual PP was used through GitHub to allow both programmers to work collaboratively. Implementing a PP environment showed that mistakes were drastically reduced as both programmers could correct each other. Not only this but programmers are able of sharing each other best practices improving the approach and increasing the speed. The attributes of PP meant that the artifact could be created swiftly and correctly throughout all sprints.

Test Driven Development:

“Test-Driven Development is the evolutionary approach in which unit test cases are incrementally written prior to code implementation.” (Kumar & Bansal, 2013)

The sequence is as followed for a TDD:

Add a Test, Run Validation Test, Write Code, Run Tests, Refactor Code.

The TDD is a model where a test is created at the beginning and then used to test code at the end. This helps to define and design code so that task revolves around passing the test, which increases the application build time and quality. However, throughout the artifacts creation team members noted that the designing tests took a considerable amount longer than standard convention of testing at the end or creation.

Velocity:

A requirement of Velocity is it must monitor and predict PV, considering the impact of relevant process factors. Also “Roughly speaking, PV can be thought of as a productive effort per iteration.” (Hearty, et al., 2009). This is seen to be used in the product owners backlog as it mentions what’s story points are for each task and the prioritization of the epic user stories.

Estimation:

The framework allows tasks and completion times to assigned based on what the team believes they will need. These are discusses and decide during the sprint meeting and assigned to an individual sprint. These sprints which start and end on specific dates. The length is decided during sprint meetings which include the product owner, Scrum master and Team. “Scrum uses fixed-length iterations... which are typically two weeks or 30 days long” (Sultana, 2015).

CLASS & RELATIONSHIP DIAGRAMS AND DFD’S

This is a layout explanation for the ‘CLASS & RELATIONSHIP DIAGRAMS AND DFD’S DOCUMENT’ which has been submitted in supporting documents. It provides a structure of the document and what you’ll find within it. The structure works as followed: sprint number, UML, class and

Testing, UML Diagrams and Paired Programming from Sprint 4:

User Stories:

- As a user I want a search result GUI display for both OMDb and TMDb so that I can see the returned movie.
- As a user I want to display all Wishlist items on a GUI screen so that I can see all the movies I intend on watching.
- New: As a user I want to be able to search and delete from the wish list.
- As a use I want to display all Previously search movies on GUI display so that I can visit other films I may have forgotten about.
- New: As a user I want to be able to search, delete and add to the wish list from the search logs because I may forget.

Class Diagram:

Class Diagram including two design patterns can be found on the next page. More diagrams can be found in the supporting documents.

Observer Design Pattern:

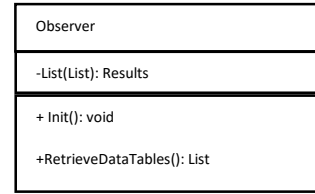
The Observer Pattern has its own Class to pull constant information from the API to update all GUI Elements needing Database Table data.

Singleton Design Pattern:

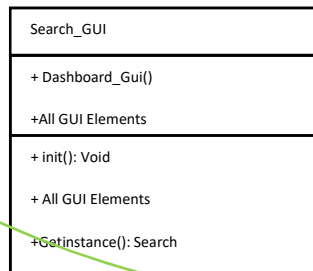
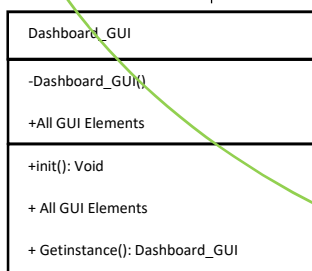
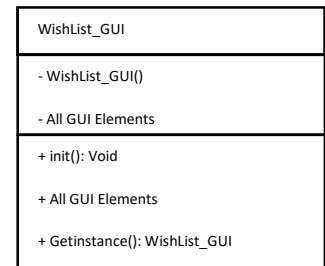
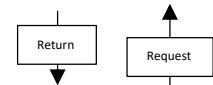
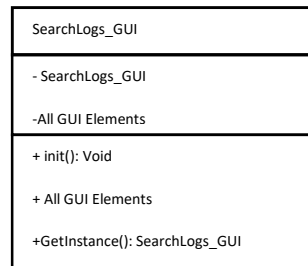
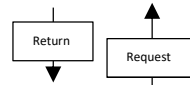
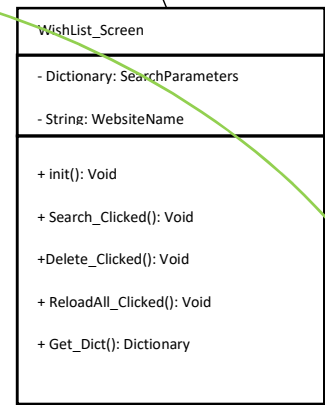
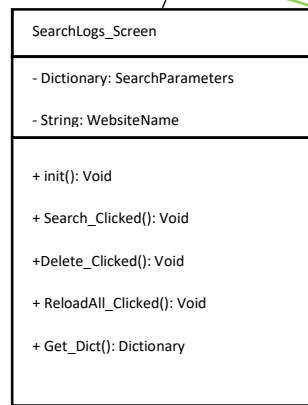
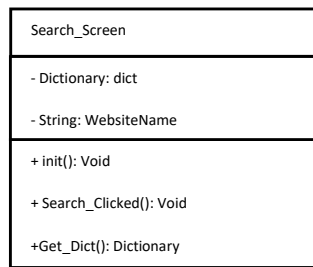
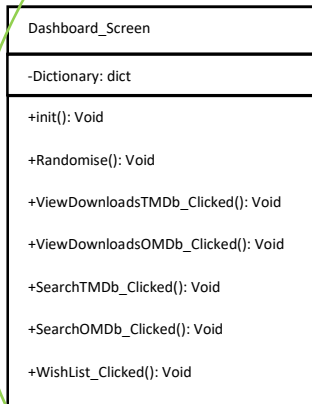
The Singleton Pattern enables a single instantiation of a class.



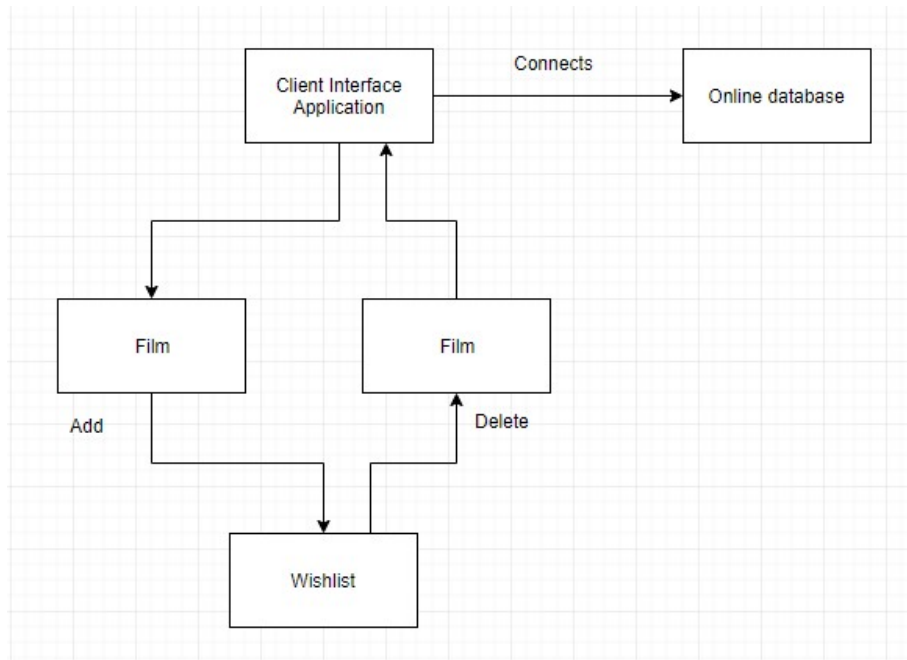
Observer
Design Pattern



Singleton
Design Pattern



Domain Model:



Test Driven Development:

The below represents one of the TDD designs more can be found in supporting documents. This table represents the dashboard GUI's Randomise Method.

Dashboard TDD:

Randomize Method:

Test	Passed/Failed	Implement Code	Passed/Failed
Randomise Method	Failed	Implement single Randomise Method	Passed
IMDbID	Failed	Implement IMDbID	Passed
IMDbID == "tt"	Failed	Implement check equal to "tt"	Passed
Length IMDbID == 7	Failed	Implement for Loop to add random numbers to the IMDbID	Passed
ParameterList	Failed	Implement ParameterList	Passed
ParameterList == 7	Failed	Implement ParameterList with the 7 elements	Passed
WebsiteInfo	Failed	Implement WebsiteInfo	Passed
DataDictionary	Failed	Implement DataDictionary	Passed
Check Dictionary == 14	Failed	Convert DataDictionary to list	Passed
SearchResult	Failed	Implement Class variable	Passed

Paired Programming Logs:

Below represent two of the Programming logs used in sprint 4; more can be found in supporting documents.

Sprint 4; PP 1		
Date & Time	Monday 19th November 2018, 1pm - 3pm	
Team	Driver	Watcher
	Team Member 3	Brandon Slater
Activity (what's being coded, tested or compiled)	<ul style="list-style-type: none">• Link the Dashboard Wishlist button to this gui so that when clicked, it opens the wish list.• Link to the wish list module and display results into a table.	
Problems	<ul style="list-style-type: none">• Limited Number of Comments provide in code• Low productivity due to off topic discussions.	
Comments	<ul style="list-style-type: none">• Program was functional, and with the code being ungraded, more effort was put into the logs.	
Sprint 4; PP 2		
Date & Time	Monday 19th November 2018, 1pm - 3pm	
Team	Driver	Watcher
	Team Member 4	Team Member 2
Activity (what's being coded, tested or compiled)	<ul style="list-style-type: none">• Link to the search log class to the gui so that either the OMDb or TMDb search logs from the database tables can be displayed into a table.	
Problems	<ul style="list-style-type: none">• Linkage to GUI found troublesome, through previous coding techniques• Limited Number of Comments provide in code	
Comments	<ul style="list-style-type: none">• Program was functional and had the implemented task	

Critical Reflection on the Development of the ACME Client using SCRUM

Implementing a SCRUM and its Processes

Scrum:

A Scrum usually includes a “Product Owner...Identify product features...ScrumMaster is responsible for ensuring the Scrum process is followed...Development team a self Organising group of software developers” (Sommerville, 2016). Scrum is a method created outside the conventional project management perspective and uses names which have replaced a normal project title. “The developers of scrum wish to make clear that Scrum was not a method for project management” (Sommerville, 2016). A scrum is the preferred agile process to create small pieces of software in a short time. Using a Scrum for large projects requires changes to the structure to be effective. The ACME movie client development team consists of 4 people which is very beneficial for this type of agile process because its tailored to suit smaller groups. “Scrum is a lightweight process framework for agile development and was...designed for smaller projects with limited team size” (Sultana, 2015).

Product Backlog:

A product backlog is used to split the product owner’s ideas into tasks that need completing during a sprint. “This is a list of ‘To do’ items that the Scrum team must tackle” (Sultana, 2015) The backlog provided a great way to interact with the product owner and share his/stakeholders ideas with the team, which strengthened communication in the hierarchy. “to identify product features or requirements...prioritize these for developments” (Sommerville, 2016). The benefit of using this backlog was that the PO had already prioritized the ideas, which meant the team knew the most important aspects to break down during the sprint planning. This “Prioritisation provides direction to the team, and it supports sprint planning”. (Pichler, 2013). The use of estimation is used throughout all iterations of the product backlog including story points and business values. These values are inherently inaccurate and imprecise, which can make them hard to decipher. (Gilley, 2014). However, the use of these values compliments the flexibility of the agile method whilst also providing specific rationale.

Sprint Planning:

Throughout the sprint meetings the product owner provided new “set of development activities which are undertaken during a pre-determined timeframe” (Ionel, n.d.). Along with edited appraisals on what we were doing to complete the backlog. Without this feedback and updated backlog, the team wouldn’t know how to proceed, and the project would take longer due to more iterations being needed. Overall the meetings were productive, and we gained sufficient information from the product owner, but some of his ideas were simplistic and had no prior influence. This meant that the team didn’t have 100% transparency and designed GUI’s to their own preference. In our case this was acceptable to the owner but for future sprint meetings the PO needs to be more involved and knowledgeable on what they want. Without the knowledge on what designs the customer will want, the application could have the aesthetical appearance; leading to increased product development time through iterations. The use of estimation has been implemented across Sprint planning process to allow judgements on time (Rogers, 2017). The most commonly used estimation is time, where for each task in the sprint backlog a specified number of hours needed agreed upon. Another is the story points estimation model “which consists of three phases the story size estimation, velocity optimization, and effort estimation” (Raslan, et al., 2015).

Sprint Backlog:

The importance of a sprint backlog was that it provided detailed user stories from the product backlog. (S.Jansi & Mrs.K.C.Rajeswari, 2015). A benefit of incorporating detailed user stories is that it shows more knowledge and technical steps needed to accomplish them. (Rubin, 2013). Using the XP process velocity, provided useful for the estimation of dedicated hours per task. However, these are only considered estimates as no calculations are present to provide accurate completion times. This can be considered a disadvantage because actual completion times could be shorter or longer dependent on the task.

Daily Scrum Meeting; x4 days a week:

The daily standup meeting was a good approach to report on the previous day's activities. (Permana & BALI, 2015). The meeting "By addressing these questions, the most important aspects of development will be covered each day" which gave motivation through strict hours of completion. (lonel, n.d.) These meetings allowed team members to be more involved in project decisions. This involvement motivated team members who saw progress in their work. (lonel, n.d.) However, group members who weren't meeting their daily objectives noticed more stress and were less motivated when they had the previous day's work to complete. (lonel, n.d.)

Sprint Review:

"The review covers the functional, executable systems, which incorporate the objects associated to the respective team, and includes the changes made to implement the journal entries." (lonel, n.d.). During the sprint review our application was show cased to the client and any stakeholders involved during the backlog. This time we were able to receive feedback from the product owner and get his opinion on how the application was developing. (Romano & Silva, 2015). A benefit of the review was the time spent with the client; throughout the meeting we able to ask questions, which provoked the PO to increment updates to certain aspects of the sprint. However, without the review of the previous sprint; getting valuable knowledge from the product owner would be challenging. This may lead to decreased motivation due to the lack of appraisals or correctness confirmations. (Romano & Silva, 2015).

Sprint Retrospective:

"This activity frequently occurs after the sprint review and before the next sprint planning." (Rubin, 2013). The retrospective is a great way to adapt the process that has led the team to its current state. This allows all teams members to discuss what could be done better during the next sprint, which in turn provide better results in the long term. (Rubin, 2013). This complements the SCRUM methodology as it provides continuous improvements for involvement, communication and the processes used within the sprint. Without this, team members won't reflect on the bad aspects of the last sprint, which could be taken over into the next, causing delays and more development iterations.

XP Processes:

User Stories:

The stories enabled our team to understand what the product owner wanted; allowing the development to primarily be based on the customer's needs. This boosts transparency between the team and the PO; strengthening trust and increasing development. (Patel & Ramachandran, 2009). Also, the information from the product backlog is detailed enough to provide ample knowledge to make subtasks. However, this may not be the case in all SCRUM's. The user stories can be simplistic, leading to less detailed subtasks and increase iterations. Furthermore, user stories allow times to be identified, allowing effective organization across the sprint. (Hearty, et al., 2008)

Paired Programming:

Implementation of the paired programming method increased productivity and allowed our team to write code and analyse code simultaneously. Also, team members were found to be more comfortable and satisfied by working in pairs. (Cockburn & Williams, n.d.). Moreover, team members started to communicate on unrelated topics, leading to distractions and increased time spent per iteration. However, research found that employing two people only had a small developmental cost but resulted in fewer defects, which could ultimately reduce offset costs. (Cockburn & Williams, n.d.). PP doesn't work alongside typical programming conventions. Programmers tend to have reluctances in sharing and coordinating working times & version controls with another person. On the other hand, problem solving and learning are both qualities found when incorporating paired programming, providing evidence that two is better than one. (Cockburn & Williams, n.d.).

Test Driven Development:

TDD has been implemented into the designing and writing of our groups python application. Its use inherently to test code before its written to ensure Quality is improvement by fixing errors at the starting point of development. (Ahsan & Khanam, 2017). On the other hand, writing test cases doesn't guarantee an error free application but has a "42% lower defect density" (Ahsan & Khanam, 2017). Also, implementation of test cases can increase the amount of time spent on a single method rather than designing and adding new functionality. In comparison to Unix testing, TDD developers improved their speed as more time was spent debugging; however, a lot time and effort are required to design test cases. (Ahsan & Khanam, 2017).

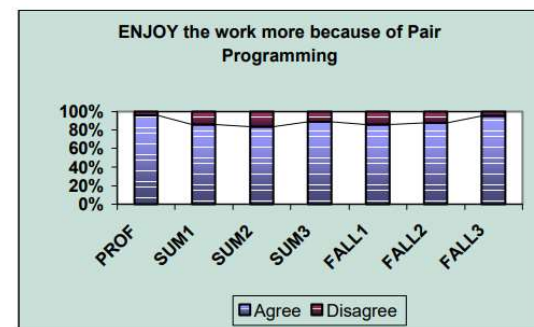


Figure 3: Pair Enjoyment

Comparison of SCRUM vs Other None Agile and Agile Methodologies

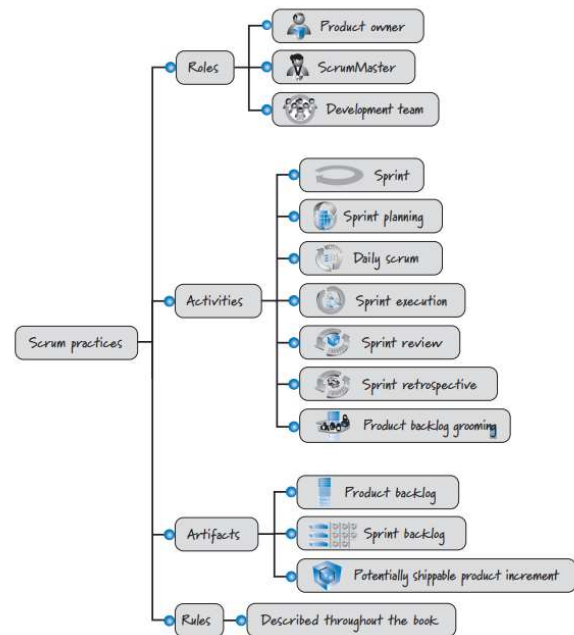
Agile Processes:

SCRUM

A SCRUM allows for a small project team to work across a project iteratively. This small size can be considered a weakness by not having the correct amount of team members to carry out the project. (lonel, n.d.). Our team worked better in a small team because it allowed us to work more closely and control work flow better.

Another impact is if the project is carried through external project developer. This is due to the PO not being around to communicate with internally, which in some projects can provide difficult if appropriate knowledge isn't gained. (lonel, n.d.). This was not an issue in our project development as our PO was very involved during his reviews and meetings. However, the SCRUM meetings were very beneficial as they provided our team with time to discuss ideas with the PO and helped organize our team better. This impact affects all models as you can never guarantee than the user will be productive in their communication.

On the other hand, an impact of the SCRUM model is the long period to which the user cannot intervene. (lonel, n.d.). This provided challenging as new ideas had to wait till the end of the sprint to be implemented; having flexibility is beneficial but not during sprints. This is different to the other models as they can change throughout ensuring complete transparency with the user.



Dynamic System Development Method (DSDM)

DSDM is used for quick product delivery through Rapid Application Development. RAD can be described as “the commercial need to deliver working business applications in shorter timescales and for less investment” (Beynon-Davies, et al., 1999). This method uses the following stages for development; Feasibility Study, Business Study, Functional Model Iteration, Build & Design Iteration, and Implementation. This is like the SCRUM model as it also has aspects which are iterated, but its main feature is design and rapidly producing a prototype to fulfil the user (owners) requirements. (Shelly, 2015). This would be beneficial to the ACME development as many prototypes can be created to meet the requirements and is iterated till the user is satisfied. (Al-Saleem & Ullah, 2015).

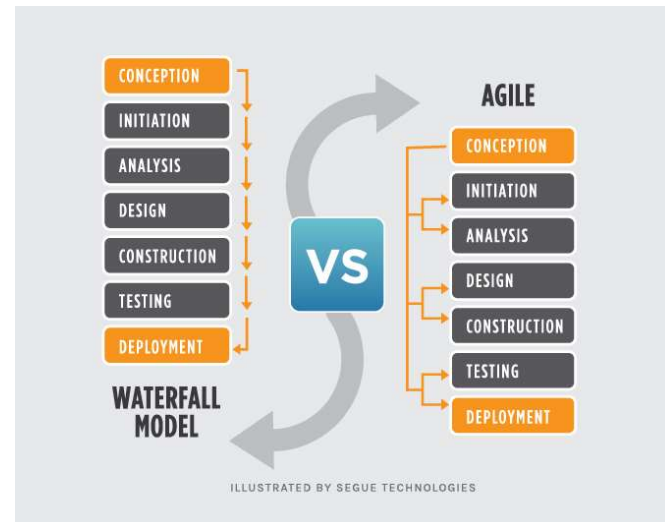
This model is orientated to minimize the number of risks through the iterative approach. (Al-Saleem & Ullah, 2015). Like all models but has a step to reduce risks through analysis and studies, which helps remove vulnerabilities that could affect the application. DSDM is like scrum in relation to the prototypes created; however, SCRUM looks to have a deliverable application whereas the DSDM is still only a prototype.

However, DSDM has more user involvement than the SCRUM model, which has benefits and drawbacks. Benefits being that the active user involvement is important to maintain the user-centered approach

DSDM implements. (Beynon-Davies, et al., 1999). This could improve the ACME development as our team could spend more time with the user understanding their needs and criteria, instead of using the product backlog provided in a SCRUM. However, “More user involvement can be danger some time if the user is not an appropriate one.” (Al-Saleem & Ullah, 2015). This would be disadvantageous to the app development because the user may get too involved inceaseing the production time, leading to more iterations.

Waterfall

The waterfall model primarily works around the development of an artefact through sequential stepping. This is used to ensure that all the designs are flawless and is usually implemented where quality control is a major concern. (Alshamrani & Bahattab, 2015). Using the linear completion sequence tasks do not overlap, ensuring that one stage is completed before the next. This model has drawbacks for the creation of the ACME application. Team members would all have to work on the same task before moving onto the next, which increases the length of an applications development. However, if this project needed a higher level of security, implementing tasks this way would reduce vulnerabilities through extensive designing. (Alshamrani & Bahattab, 2015).



Designing and implementing error detection are important aspects of the waterfall method, providing a high level of technical documentation and error capturing before moving onto the next stage. (Kannan, et al., 2014). These aspects would be beneficial to the ACME development because it ensures the application wont crash to unexpected input parameters. However, all the designing and documentation is completed at the start of the application development, which doesn't consider environmental changes that may need to be implemented. (Kannan, et al., 2014). The SCRUM method has a better structure in relation to its designing and documentation through regular meeting and product owner interventions.

Whereas with SCRUM, Spiral, DSDM and other agile methods provide the user with a product to view and evaluate. The waterfall follows its sequential order and produces an application at the end of the life cycle, which doesn't accommodate changes easily. (Kannan, et al., 2014). This sequential stepping isn't beneficial for teams as it doesn't allow for multiple stages to be attempted at once. However, this can be considered an advantage to ensure there are not mistake during the current stage before moving onto the next.

Spiral

The spiral model is like the waterfall method as it looks to reduce risks by including a risk analysis stage in which a project is broken down into segments, used for ease-of-development. (Alshamrani & Bahattab, 2015). This model is also like the SCRUM in the way it goes through an iteration. Each spiral goes through a planning, risk analysis, development, evaluation phase, and repeat. The similarity to

SCRUM is seen in the many iterations of its spiral phases followed by an evaluation of the previous steps. Again, this method looks for reducing vulnerabilities in the application and can be useful in the ACME example if this was a requirement set by the Project Owner.

An impact of Spiral is the high costs associated with the increased amount of documentation and analysis. (Alshamrani & Bahattab, 2015). This impact can also be considered a benefit due to the strong documentation control and risk analysis, ensuring app confidence. However, this model is commonly used for projects that need to ensure integrity constantly and wouldn't be suitable for ACME development due to simplicity and minimum risk. (Alshamrani & Bahattab, 2015)

The spiral model provides early life prototyping useful for user evaluation and feedback which can be used to increment another spiral, enabling further advancements. (Alshamrani & Bahattab, 2015).

Evaluation of Software Tools

GitHub & Git

What is GitHub & Git?

GitHub is a website used for hosting and sharing remote git repositories. The benefit of using GitHub is that its easily accessible from remote locations, allowing collaboration between non-local team members. It allows uploads of complete & uncomplete code and source software, which are most commonly used for sharing and allowing open source development. (LibraryCarpentry, 2016). GitHub is accessible through free and paid memberships, providing private and public labeling for the accessibility of your project. The free version doesn't allow privatization of projects, meaning, that all GitHub users can access and download your repositories. This is known as open source software and be seen as a disadvantage in using it.

Git on the other hand is a popular version control system which allowed our team members to access the working artefact at any time to the see the development. This enabled our team to work remotely whilst ensuring access to the current working program. The major benefit was the ability its version control system provided my team throughout the development of the ACME application. (Perez-Riverol, et al., 2016).

However, the complexity of the git repositories provided competency challenges in the beginning, leading to an increase in time for training. (Clancy, 2016). Although it was complex and difficult to use at first, the overall abilities that it provides are immense. One heavily used git feature was branching. Branching allowed the team to work simultaneously through creating copies of the main branch and implementing their codes onto that version. This provided members with a safe environment to work on unstable code which wouldn't affect the rest of the application. (Perez-Riverol, et al., 2016). Without checking the compatibility of the commit, the integrity of the main branch may be destroyed through poorly written and tested code. However, considering this is a web-based application the user is highly dependent on internet access to collaborate remotely. This wasn't an issue during the ACME application development as we worked locally, and issues were resolved through local version control. This could also affect the commits because members weren't able to merge their code with the main branch. (Ricketts, et al., 2013)

GitHub is a useful tool for multiple team members to work across, as it provides many tools for cross development. Permissions are another setting used to grant a user access to specific parts of the repository. This is beneficial for restricting public access to the project, which brings weaknesses. (Chacon & Straub, n.d.). The project was made private to ensure that public users wouldn't have access to the team's project. However, the repository wasn't created to share with public GitHub users, as the code is written to only be compatible with specific software specifications. This meant that all members had to have the same software builds to ensure compatibility when cloning and committing changes. Without this, my group couldn't work remotely and have the same organized version control that GitHub provides. (Ricketts, et al., 2013)

How it's use in open source software development?

GitHub is commonly used for sourcing and publishing open source code and software. Allowing code and software to be freely available allows a business's market share to grow as the community tests and grows the application. (Singh, et al., 2015). This allows free testing and development from the users who have been downloading the open source code. These communities also improve a solution meaning less error prone or more effective. Another benefit of OSS is its cost. OSS is free meaning than more of the community are likely to sample or use the app. This also reduces company costs and removes the blockade of restrictions created by large companies. (Singh, et al., 2015). However, creating a OSS can be financially restraining on the creator, but usually OSS is specific on the task its designed to do, limiting its use.

On the other hand, open source software can have many security risks. These risks may not be easily detectable because many third parties have manipulated the application over its development; Whereas, proprietary software is considered to have been created in a secure environment and tested for hackable vulnerabilities. (Singh, et al., 2015).

Another impact of OSS is the ability for programmers to steal ideas and use it as their own. There is copyleft law which prohibits abusing someone else's work to portray as your own. Whereas, proprietary software has licenses to stop the thief of intellectual property. (Singh, et al., 2015).

Furthermore, no technical support and long-term support are considered weaknesses of using OSS. (Ricketts, et al., 2013). Reduced support can be through a range of factors; the developer is busy or is no longer involved in the project. (Singh, et al., 2015). These can impact the people still using the OSS as they may need crucial updates and cannot be given them. This can implement risks and vulnerabilities in the application if not configure currently with other software versions. There is also possibilities than the software is open source and available to modify but is problematic because of the way it's written; ensuring that the technical support is billed. (Ricketts, et al., 2013)

Documentation is another problem with OSS. The implementation of a user-friendly document explaining the installation of code is crucial to ensure its usability. (Singh, et al., 2015). Without this, OSS is most likely shelved because of its complexity. This is commonly found in OSS projects because the constant updates pushed are hard to document, leading to less opportunities for modification. This does not follow the OSS convention of having a community to improve and adapt software. (Ricketts, et al., 2013)

Evidence of Use

brandontaslater / ACME_Movie_Information_Client

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Overview **Yours** Active Stale All branches

Search branches...

Your branches

DatabaseManagement	Updated 17 minutes ago by Team Member 1	0 0	New pull request	
Dashboard_GUI	Updated 17 minutes ago by Team Member 2	0 0	New pull request	
Search_GUI	Updated 17 minutes ago by Team Member 3	0 0	New pull request	

brandontaslater / ACME_Movie_Information_Client

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Overview Yours Active Stale All branches

Search branches...

Default branch

master	Updated 19 minutes ago by Brandon Slater	Default	Change default branch
--------	--	---------	-----------------------

Your branches

DatabaseManagement	Updated 19 minutes ago by Team Member 1	0 0	New pull request	
Dashboard_GUI	Updated 19 minutes ago by Team Member 2	0 0	New pull request	
Search_GUI	Updated 19 minutes ago by Team Member 3	0 0	New pull request	

Active branches

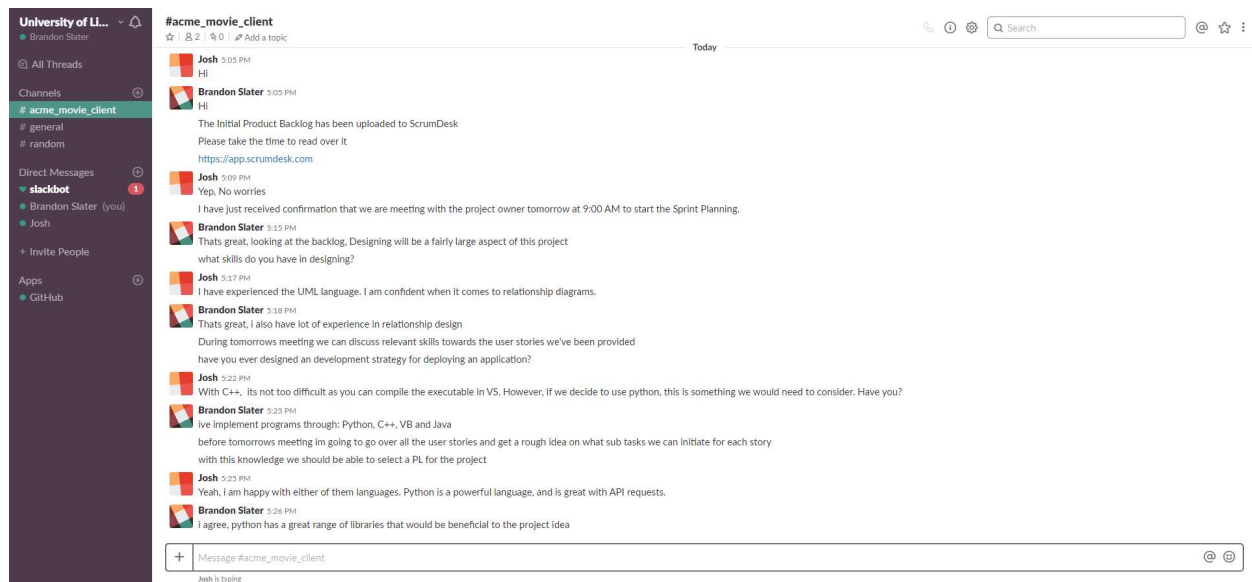
Search_GUI	Updated 19 minutes ago by Team Member 3	0 0	New pull request	
DatabaseManagement	Updated 19 minutes ago by Team Member 1	0 0	New pull request	
Dashboard_GUI	Updated 19 minutes ago by Team Member 2	0 0	New pull request	

Slack

What is Slack?

Slack is a messaging service which allows multi or single conversations within groups. Slack is accessible remotely and across a wide range of devices, allowing our team to always be in communication. Multiple channels can be established allowing different group members to communicate separately, which allowed for more productive debating and conversations. (Elorza, n.d.) This also meant that our group conversations wouldn't get cluttered and filled with meaningless chat and important discussions wouldn't get lost. However, team members weren't as responsive as they would have been through none professional methods of communication. Before using this tool, communication would be through email which wasn't as fluid and consistent, taking days to get replies to answers. (Elorza, n.d.). Furthermore, Slack provided a way to connect our most used applications, such as GitHub, Google drive and Trello. This was extremely beneficial for GitHub as it allowed simple in chat queries providing different ranges of information about the repository.

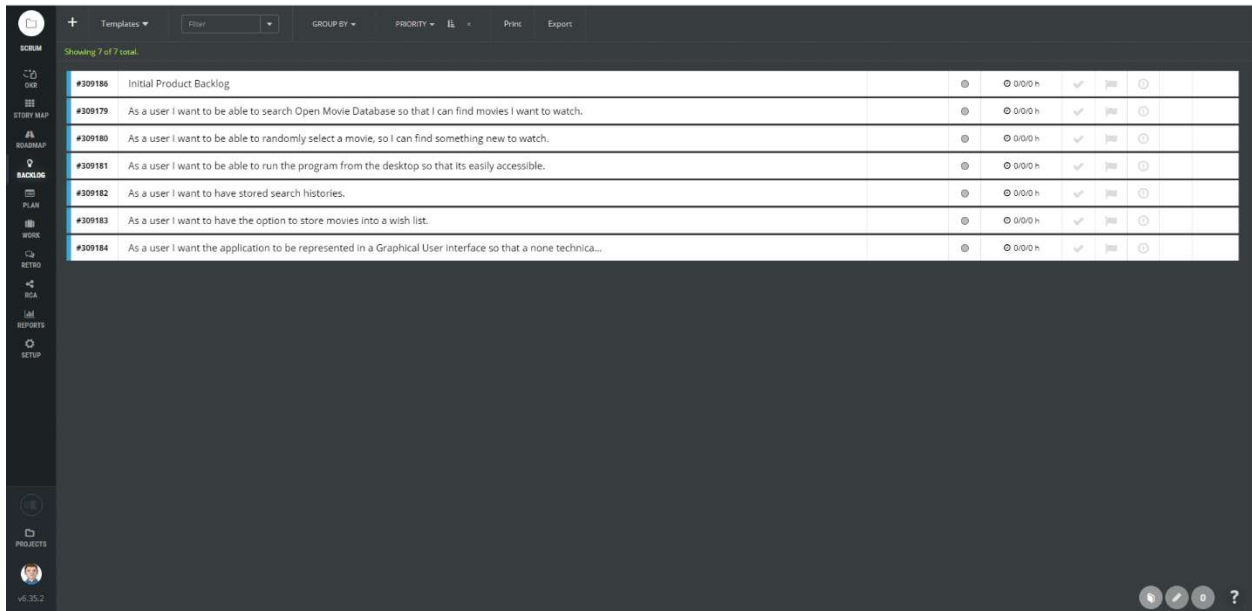
Evidence of Use



Scrum Desk:

Scrum desk is a powerful web application which provided our product owner with a way to show case his ideas into a Project backlog. This backlog allowed us to create story maps and plans on how to broaden these epic backlog ideas. The application can be remotely accessed meaning team members working from home always knew what ideas the owner had, improving communication. Along with this, the PO's communication was improved as he could add comments on his ideas to broaden what he wanted, giving the team more information to make decisions on how to implement certain aspects of the app. Without this application the involvement of the PO could have reduced leading to uncertainties on what he wanted.

Evidence of Use



The screenshot displays the Scrum Desk web application interface. On the left is a dark sidebar with navigation icons for SCRUM, OKS, STORY MAP, ROADMAP, BACKLOG, PLAN, WORK, RETRO, RGA, REPORTS, and SETUP. The main area shows a 'Showing 7 of 7 total' backlog list. At the top of the main area are controls for Templates, Filter, Group By, Priority, and buttons for Print and Export. The backlog items are as follows:

ID	Description	Progress	Estimate	Assignee	Labels
#300186	Initial Product Backlog	0/0/0 h	✓		
#300179	As a user I want to be able to search Open Movie Database so that I can find movies I want to watch.	0/0/0 h	✓		
#300180	As a user I want to be able to randomly select a movie, so I can find something new to watch.	0/0/0 h	✓		
#300181	As a user I want to be able to run the program from the desktop so that its easily accessible.	0/0/0 h	✓		
#300182	As a user I want to have stored search histories.	0/0/0 h	✓		
#300183	As a user I want to have the option to store movies into a wish list.	0/0/0 h	✓		
#300184	As a user I want the application to be represented in a Graphical User interface so that a none technica...	0/0/0 h	✓		

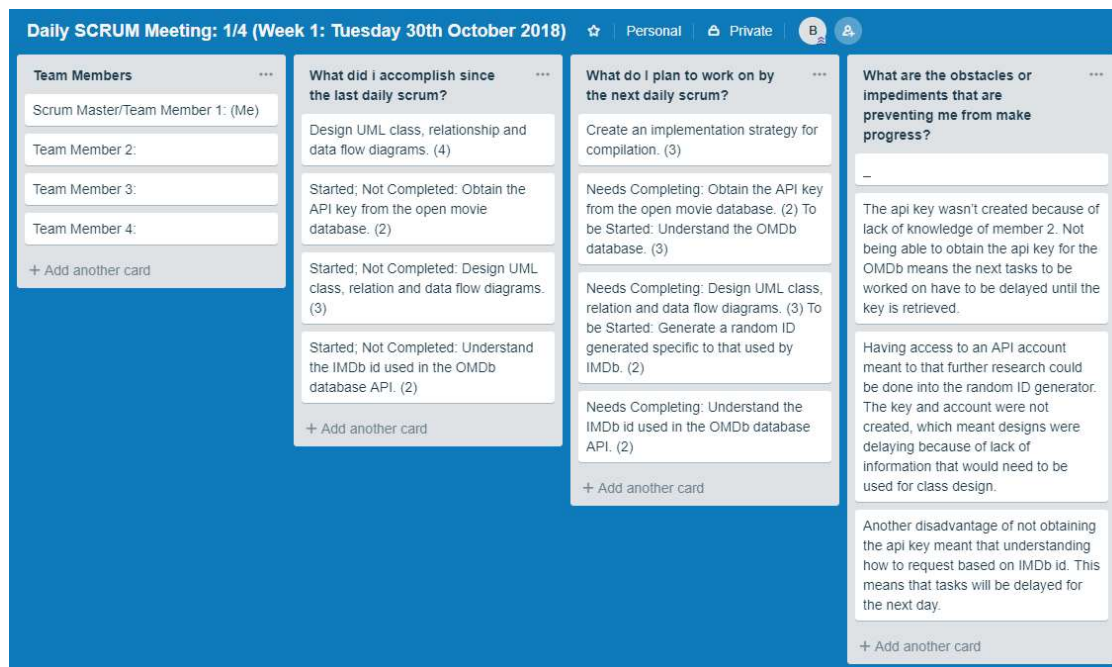
The bottom of the sidebar shows 'PROJECTS' and 'v6.35.2'. The bottom right of the main area has icons for a list, a pencil, a trash can, and a question mark.

Trello:

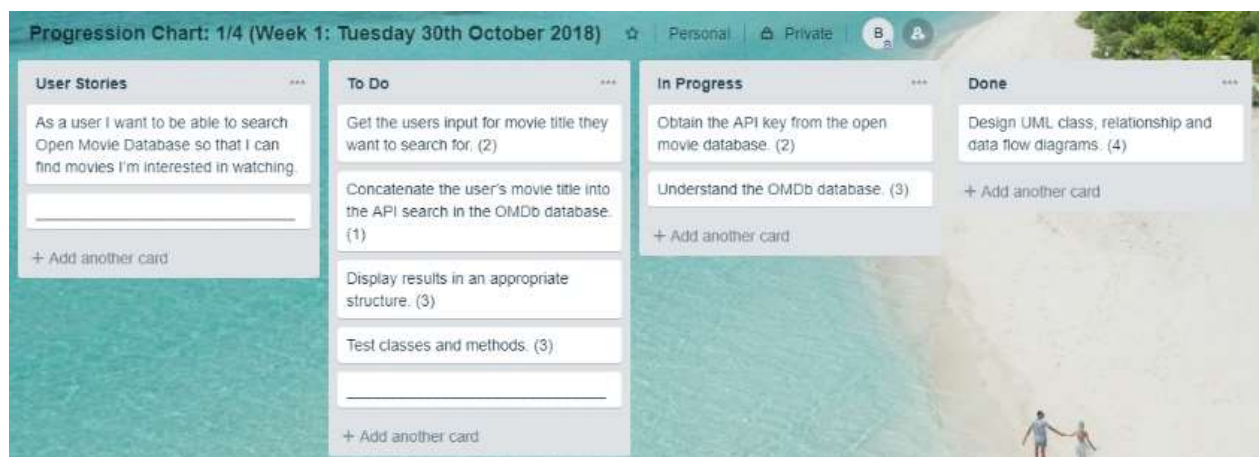
Trello is collaboration tool used to manage and organize tasks within teams. It uses “boards, lists, and cards enable you to organize and prioritize your projects in a fun, flexible and rewarding way.” (Trello, n.d.) This application provides a great way to prioritize team’s schedules and creating discussion boards for supporting the SCRUM methodology. We used Trello to store all our daily scrum meeting and progression charts. The design of Trello meant that it was easy and intuitive to drag and drop objects to new headers. Trello helped organize our project more clearly knowing what each member was doing and what tasks were left to do, this kept us all motivated to do work.

Evidence of Use

Daily Scrum Meeting: 1/4 (week-1-tuesday-30th-october-2018):



Progression Chart: 1/4 (Week 1: Tuesday 30th October 2018)



PyCharm:

What is PyCharm?

PyCharm is an intelligent code editor built to provide a programming IDE capable of providing its users with complex and simplistic features able to produce code effectively and quickly. Python native was used to program the ACME application and had the following benefits. The ability to write code quickly was enforced through Intelligent Coding Assistance “smart code completion, code inspections, on-the-fly error highlighting and quick-fixes, along with automated code refactoring’s and rich navigation capabilities.” (PyCharm, 2018). The use of Version Control Systems provided the ability to segment working code and unify it into root versions once completed. “Unified support for Git...version control systems help you manage local changes and apply complex branch operations. All the tedious tasks (adding, removing and deleting files) are performed automatically.” (PyCharm, 2018).

How it’s been used throughout the development process?

Version Control System (VCS)

There are two types of version controllers used within PyCharm; Git and Local Version Control System (LVCS). These two works, “At the IDE level and at the project level” (PyCharm, 2018). The IDE level “allows you to examine the differences between two revisions of a file, or between its current local copy and the repository version.” (PyCharm, 2018). This is beneficial for error monitoring to see individual changes that the member made to a file and to test between previous code.

GitHub & Git

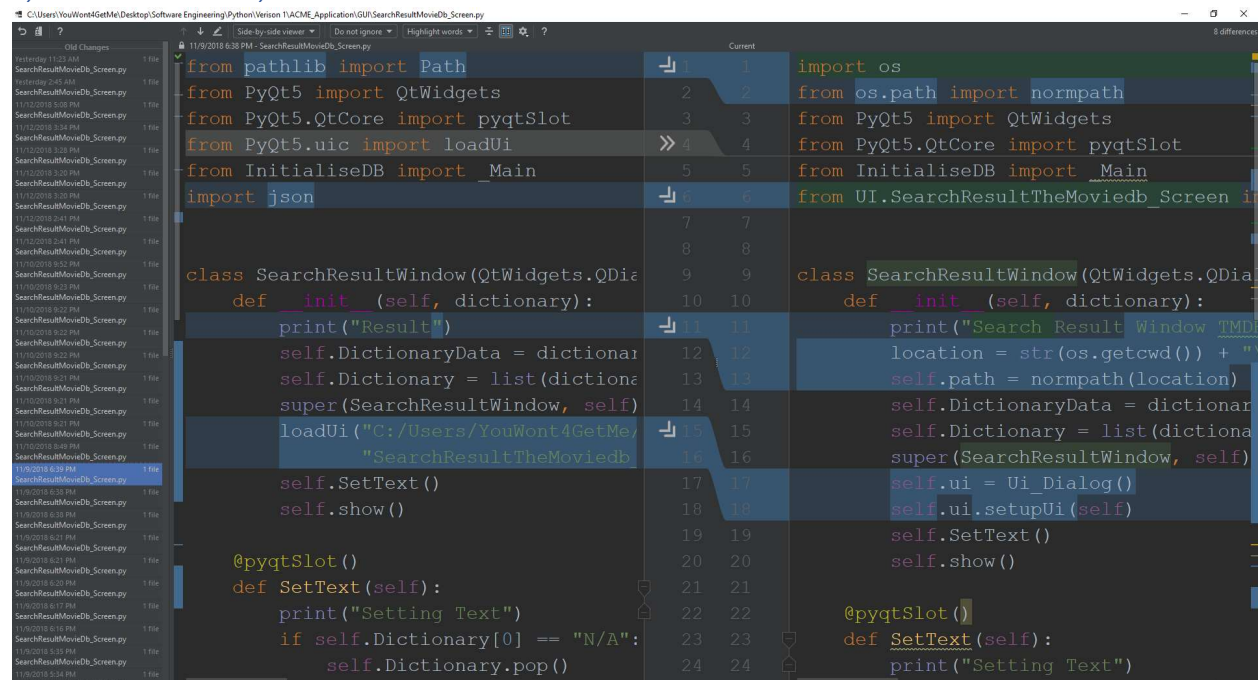
As a group we needed to be able to communicate and collaborate on program development which is an added feature within the IDE of PyCharm. This allowed our team to sync all members local repositories (Saved on their computer) with Gits remote repository; providing the following features, “Fetch Changes, Pull Changes, Commit changes, push changes, manage branches and patch” (PyCharm, 2018)

Anaconda

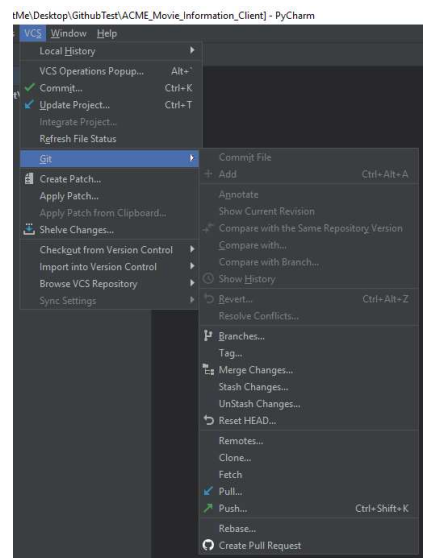
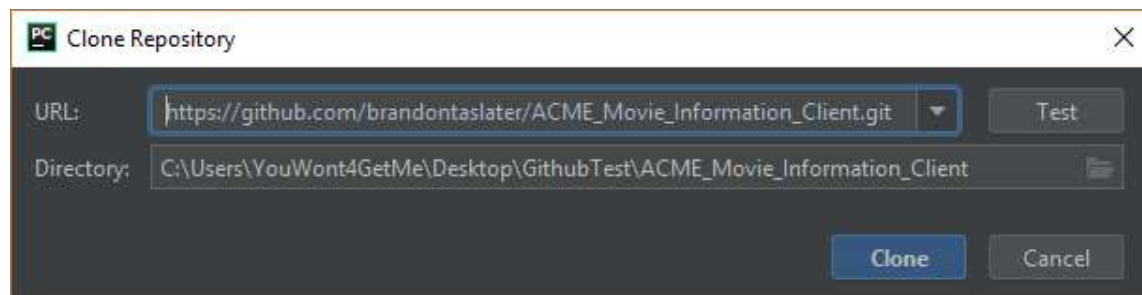
The external library installer provided beneficial in the development of the ACME application, providing a large database of different modules to advance the program. Without this piece of software installing updated libraries were difficult to find and use with current versions of Python/PyCharm. For example, two used libraries were the Database SQL3 and PYQT3. These libraries had to be the same on all team members computers which anaconda provided a solution too; packaging all modules into one configuration folder allowing multi use through file transfer. This ensured all members were running the same versions throughout programming on their local computer.

Evidence of Use

Pycharm Local History



Pycharm VCS using Github



References

- Ahsan, M. N. & Khanam, Z., 2017. Evaluating the Effectiveness of Test Driven Development: Advantages and Pitfalls. *International Journal of Applied Engineering Research*, 12(18), pp. 7705-7716.
- Al-Saleem, S. M. & Ullah, H., 2015. A Comparative Analysis and Evaluation of Different Agile Software Development Methodologies. *IJCSNS International Journal of Computer Science and Network Security*, 15(7), pp. 39-45.
- Alshamrani, A. & Bahattab, A., 2015. A Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model. *IJCSI International Journal of Computer Science Issues*, 12(1), pp. 106-111.
- Beynon-Davies, P., Carne, C., Mackay, H. & Tudhope, D., 1999. Rapid application development (RAD): an empirical review. *European Journal of Information Systems*, 8(3), pp. 211-223.
- Bryant, S., Boulay, B. d. & Romero, P., 2006. *XP and pair programming practices*, s.l.: Research Gate.
- Chacon, S. & Straub, B., n.d. *Pro Git*. Second ed. s.l.:Apress.
- Clancy, J., 2016. *CodeClouds*. [Online]
Available at: <https://www.codeclouds.com/blog/advantages-disadvantages-using-github/>
[Accessed 30 11 2018].
- Cockburn, A. & Williams, L., n.d. *collaboration*. [Online]
Available at: <https://collaboration.csc.ncsu.edu/laurie/Papers/XPSardinia.PDF>
[Accessed 30 11 2018].
- Elorza, C., n.d. *Internationa House Journal*. [Online]
Available at: <http://ihjournal.com/experiments-with-slack-team-messaging-in-training-by-cecilia-elorza>
[Accessed 30 11 2018].
- Gilley, C., 2014. *Quora*. [Online]
Available at: <https://www.quora.com/In-sprint-planning-what-are-the-pros-and-cons-of-using-story-points-hours-or-both-when-creating-estimates>
[Accessed 30 11 2018].
- Hearty, P., Fenton, N., Marquez, D. & Neil, M., 2008. Predicting Project Velocity in XP Using a Learning Dynamic Bayesian Network Model. *IEEE Transactions on Software Engineering*, 35(1), pp. 124-137.
- Hearty, P., Fenton, N., Marquez, D. & Neil, M., 2009. Predicting Project Velocity in XP Using a Learning Dynamic Bayesian Network Model. *IEEE Transactions on Software Engineering*, 35(1).
- Ionel, N., n.d. *CRITICAL ANALYSYS OF THE SCRUM PROJECT*, Buchares: The Academy of Economic Studies Buchares.
- Kannan, V., Jhajharia, S. & Verma, D., 2014. Agile vs waterfall: A Comparative Analysis. *International Journal of Science, Engineering and Technology Research*, 3(10), pp. 2680-2686.
- Kumar, S. & Bansal, S., 2013. Comparative Study of Test Driven Development with Traditional Techniques. *International Journal of Soft Computing and Engineering (IJSCE)*, 3(1), pp. 325-360.

- LibraryCarpentry, 2016. *Library Carpentry*. [Online]
Available at: <https://librarycarpentry.org/lc-git/01-what-is-git/>
[Accessed 30 11 2018].
- Patel, C. & Ramachandran, M., 2009. Story Card Based Agile Software Development. *International Journal of Hybrid Information Technology* , 2(2), pp. 125-140.
- Patel, C. & Ramachandran, M., 2009. Story Card Based Agile Software Development. *International Journal of Hybrid Information Technology* , 2(2), pp. 125-140.
- Perez-Riverol, Y. et al., 2016. *Ten Simple Rules for Taking Advantage of Git*, UNITED STATES: PLOS COMPUTATIONAL BIOLOGY.
- Permana, P. A. G. & BALI, S. S., 2015. Scrum Method Implementation in a Software. (*IJACSA*) *International Journal of Advanced Computer Science and Applications*, 6(9), pp. 198-204.
- Permana, P. A. G. & BALI, S. S., 2015. Scrum Method Implementation in a Software Development Project Management. (*IJACSA*) *International Journal of Advanced Computer Science and Applications*, 6(9), pp. 198-204.
- Pichler, R., 2013. *romanpichler*. [Online]
Available at: <https://www.romanpichler.com/blog/the-product-backlogs-strengths-and-limitations/>
[Accessed 18 11 2018].
- PyCharm, 2018. *PyCharm*. [Online]
Available at: https://www.jetbrains.com/pycharm/features/coding_assistance.html
[Accessed 30 11 2018].
- Raslan, A. T., Darwish, N. R. & Hefny, H. A., 2015. *Effort Estimation in Agile Software Projects using Fuzzy Logic and Story Points*. Cairo , Cairo University.
- Ricketts, I., Hanso, V. L. & Heron, M., 2013. *Open source and accessibility: advantages and limitations*, s.l.: Springer.
- Rogers, P., 2017. *Better Together — XP and Scrum*. [Online]
Available at: <https://medium.com/agile-outside-the-box/better-together-xp-and-scrum-c69bf9bffcff>
[Accessed 30 11 2018].
- Romano, B. L. & Silva, A. D. d., 2015. *Project Management Using the Scrum Agile Method: A Case Study within a Small Enterprise*. Las Vegas, IEEE.
- Romano, B. L. & Silva, A. D. d., 2015. *Project Management Using the Scrum Agile Method: A Case Study within a Small Enterprise*. Las Vags, 2015 12th International Conference on Information Technology - New Generations.
- Rubin, K. S., 2013. *Essential Scrum*. Boston: Pearson Education.
- S.Jansi & Mrs.K.C.Rajeswari, 2015. A Greedy Heuristic Approach for Sprint Planning in. *INTERNATIONAL JOURNAL FOR TRENDS IN ENGINEERING & TECHNOLOGY*, 3(1), pp. 18-21.

Scrum.org, n.d. *SRUM.org*. [Online]

Available at: <https://www.scrum.org/resources/what-is-a-sprint-review>

[Accessed 30 11 2018].

Shelly, 2015. Comparative Analysis of Different Agile Methodologies. *International Journal of Computer Science and Information Technology Research*, 3(1), pp. 199-203.

Singh, A., R.K Bansal, P. & Jha, N., 2015. Open Source Software vs Proprietary Software. *International Journal of Computer Applications*, 114(18), pp. 26-31.

Sommerville, I., 2016. *Software Engineering*. Tenth Edition ed. Edinburgh: Pearson .

Sultana, J., 2015. *A STUDY ON APPLICABILITY OF THE SCRUM FRAMEWORK FOR LARGE*, Toronto: Ryerson University.

Trello, n.d. *Trello*. [Online]

Available at: <https://trello.com/en>

[Accessed 30 11 2018].