University of Hull

Agile Software Development

08142

Ieaun Roberts, Luke Watkins, Numesh Neal Rama, Alin Florica, Kamil Rozner (Group 11)

5-4-2018

Introduction

Our Customer the School of Engineering and Computer Science is looking into upgrading its system for quality monitoring as the current procedures in place to perform vital functions have proved to be cumbersome. What is needed is a new application that can retain all the necessary functions of its predecessor while providing a simple robust interface that requires minimal training, incorporates ease of use properties and is as reliable as the last.

One of the major issues with the current system is that the interface is too confusing to navigate, making it difficult for staff to view the necessary information. Therefore, there is an emphasis on conveying data in a simple and concise manner. Additionally, students should be able to view the database without being able to edit any of its contents. To accomplish this, a login system will verify whether or not the user is a member of staff or a university student, with different features being enabled or disabled depending on their level of access.

The created project by group 11 is a desktop application designed for our customer to be able to schedule, monitor and review requests for late submission of assignment and coursework for each relevant module.

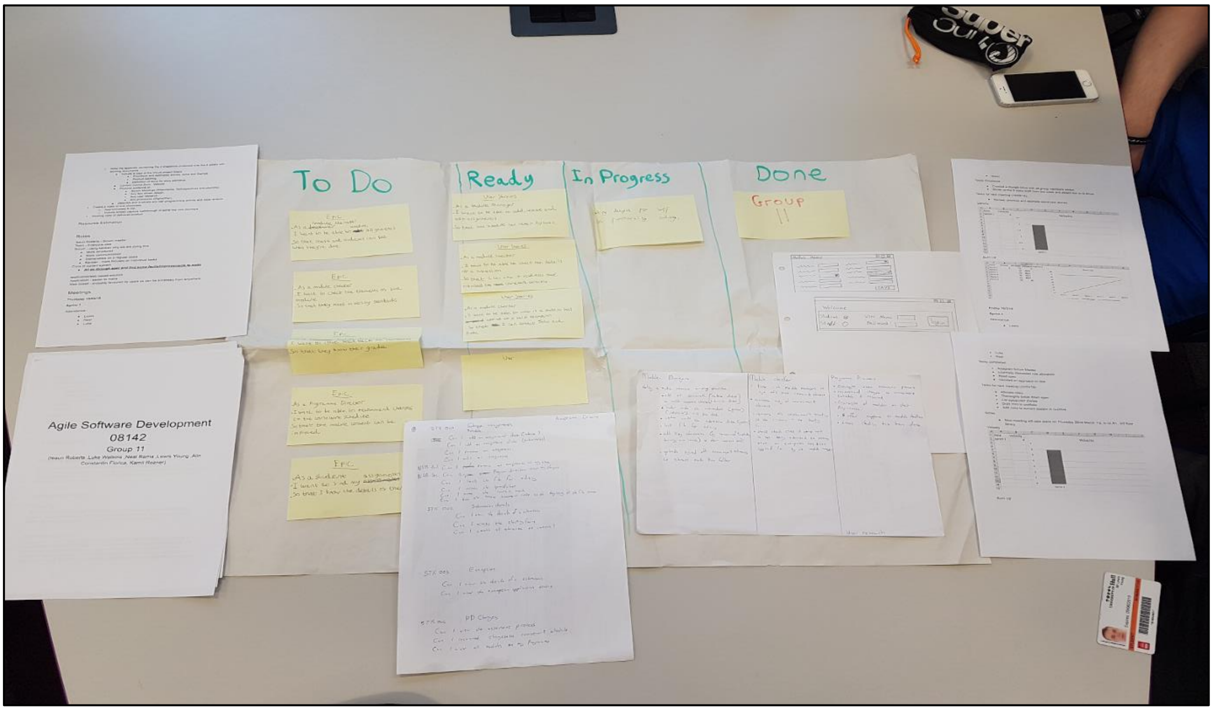
Management & Narrative

During the initial meetings of the group, roles were allocated to each group member based on their strengths and weaknesses. As none of the members in our group had any prior experience with designing and implementing a web based interface, we thought it best to play to the strengths of the team and incorporate the functionality needed into a desktop application, such a program could be used constantly without navigation to any other sites furthering the simplicity. This lead to the assignment of our 3 man development team which were tasked with prototyping and coding, Scrum master to manage both backlog and team member workload to ensure product is delivered on time and an author for the brunt of the portfolio documenting each snapshot at each checkpoint in the project.

A coherent backlog was kept and maintained throughout the lifecycle of this project. Meaning that a team member would never have been left idle at any given point in time and would always have had at least 1 task waiting to be signed off to their name. This back log was originally drafted onto a physical Kanban board then moved to a centralized location on Google drive that the team had access to and finally drafted onto Trello where it was used up till the handing in of this project.

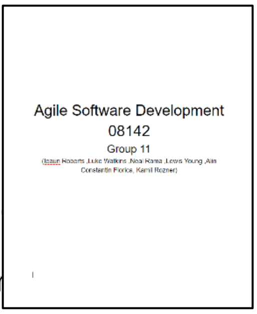
Progress throughout the project can be split up into 5 distinct segments.

Snapshot 1



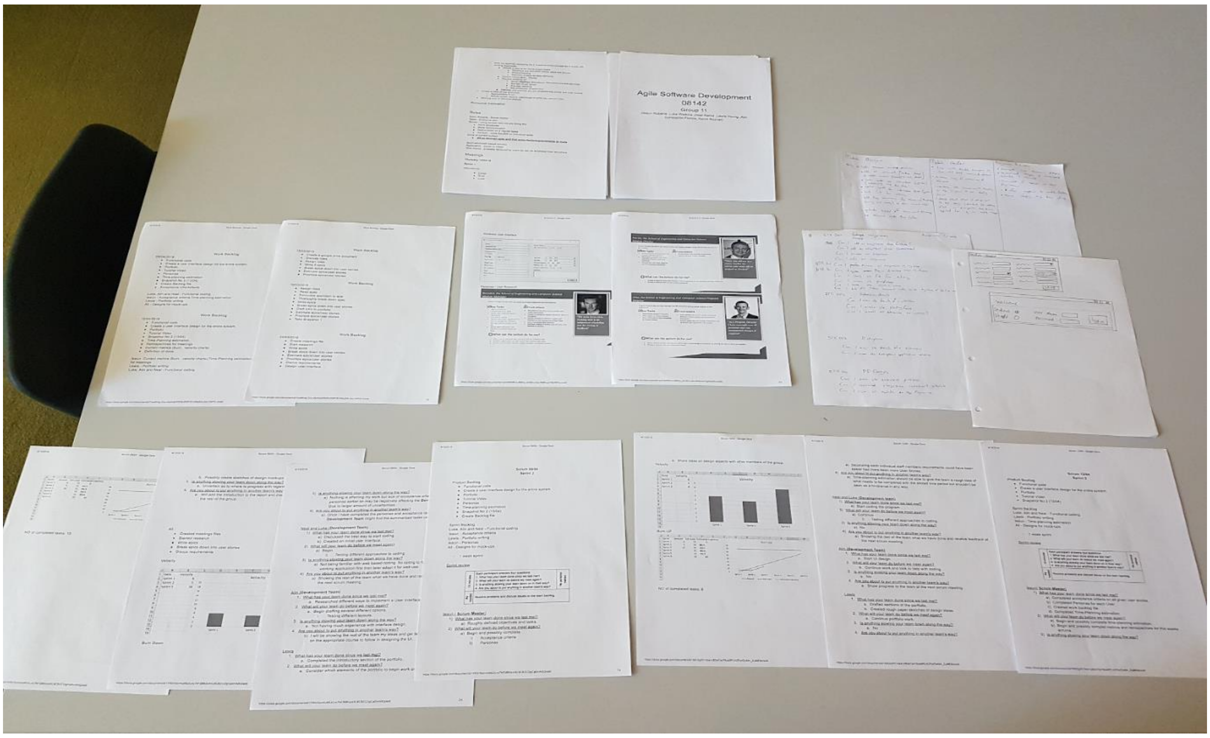
First being the initial stage (**Snapshot 1**) where a large amount of early objectives which would be related to any software development task can be found.
Namely:

- Allocation of roles.
- Initial break down of tasks into epics and users stories to be Drafted onto our user story Kanban.
- Creation of an initial planning document which contained SCRUM meeting information
- Work backlog detailing all tasks that still need to be complete
- Accpetance Test/Criteria.
- Metrics for each meeting in form of Burn up and Velocity char
- Early user research.
- Initial Prototyping to further development.



Agile Software Development
08142
Group 11
(Ieaun Roberts ,Luke Watkins ,Neal Rama ,Lewis Young ,Alin Constantin Florica, Kamil Rozner)

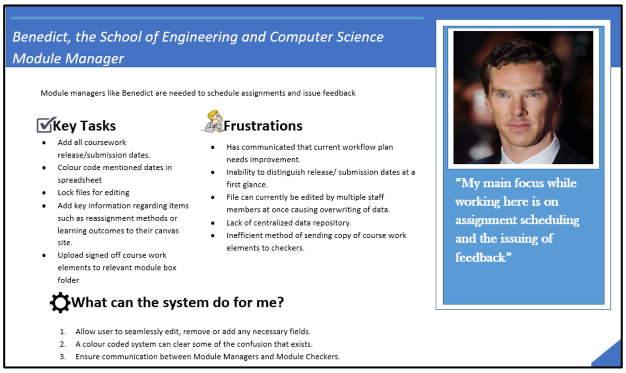
Snapshot 2



The second segment **(Snapshot 2)** can be seen as the stage in which the completion of research was attained. This was achieved through the thorough break down of requirements and non-functional requirements of each user type (Module checker, Manager, Program director). After this the Personas for the users were created in order to help the development team perceive each users essential features. Each persona contained the users key tasks, frustrations with the current system and possible features that could be implemented for the user.

With all basic functionality now documented, the development team was able to produce concise prototypes detailing the expected use of our system and role play user needs.

An estimation of time each task would take to be implemented was created to provide some sense of urgency to complete individual tasks and view the project as a whole.



Benedict, the School of Engineering and Computer Science Module Manager

Module managers like Benedict are needed to schedule assignments and issue feedback

Key Tasks

- Add all coursework release/submission dates.
- Colour code mentioned dates in spreadsheet
- Lock files for editing
- Add key information regarding items such as reassignment methods or learning outcomes to their canvas site.
- Upload signed off course work elements to relevant module box folder

Frustrations

- Has communicated that current workflow plan needs improvement.
- Inability to distinguish release/ submission dates at a first glance.
- File can currently be edited by multiple staff members at once causing overwriting of data.
- Lack of centralized data repository.
- Inefficient method of sending copy of course work elements to checkers.

What can the system do for me?

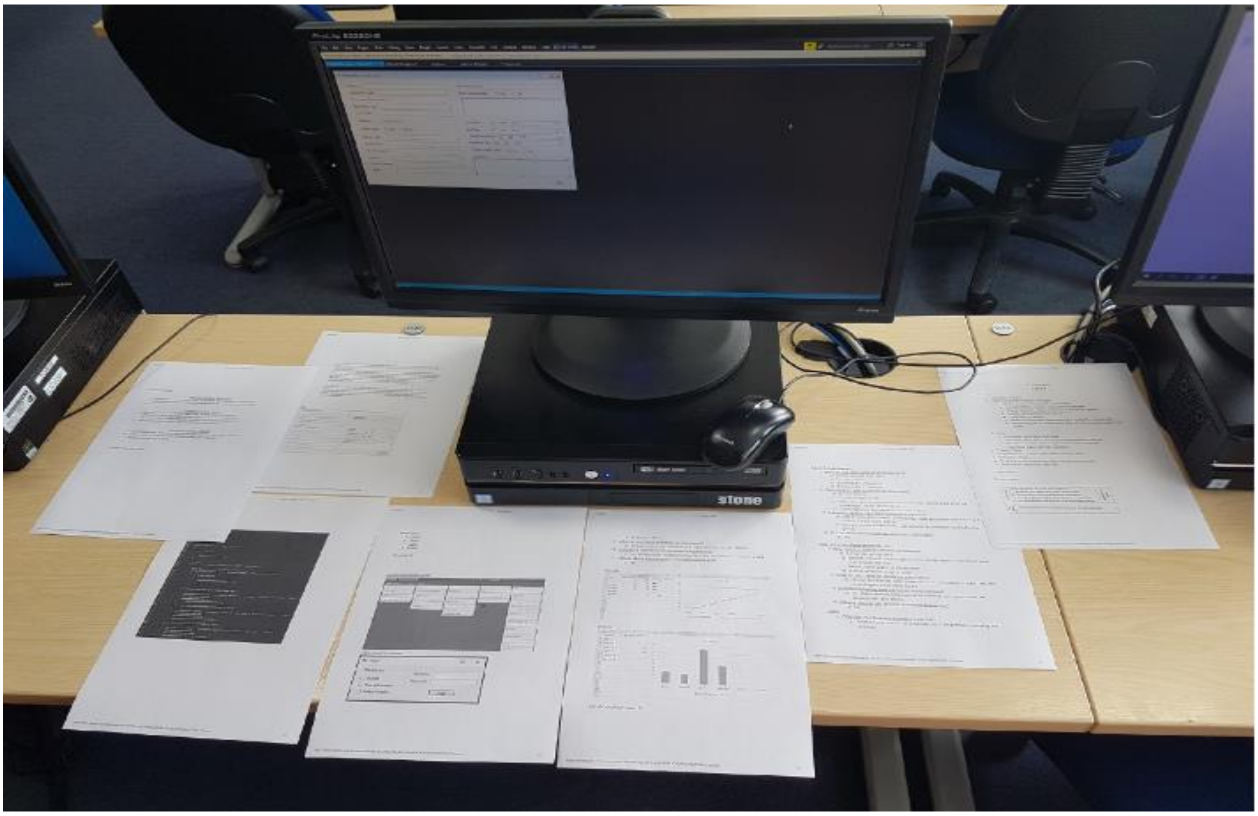
1. Allow user to seamlessly edit, remove or add any necessary fields.
2. A colour coded system can clear some of the confusion that exists.
3. Ensure communication between Module Managers and Module Checkers.

"My main focus while working here is on assignment scheduling and the issuing of feedback"

Coding began from this point relying heavily on pair programming between our now 2 man Development team. Meetings from this point on were all uploaded as individual files to google drive representing the current sprint cycle we were working in as well as the current work backlog that was ready for development. The introduction of Bi-weekly retrospective capturing at meetings meant that each member of the team could communicate what it was they had finished work on, what they planned on starting with next and also any problems they were facing along the way.

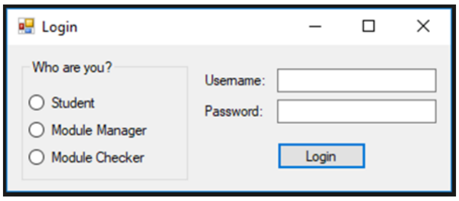
15 minutes	Each participant answers four questions: 1. What has your team done since we last met? 2. What will your team do before we meet again? 3. Is anything slowing your team down on in their way? 4. Are you about to put anything in another team's way?	No personal names
As needed	Resolve problems and discuss issues on the team backlog.	

Snapshot 3



The third segment **(Snapshot 3)** brought upon us our first coded prototype, in which some design flaws were spotted from earlier designs and promptly corrected. Pitfalls however were found in the form of SQL queries. Some task that were completed regarding the coding include

- Creation of the login screen
- Added validation to login screen
- Added functionality to the log-in screen so the system determines what class of user they are

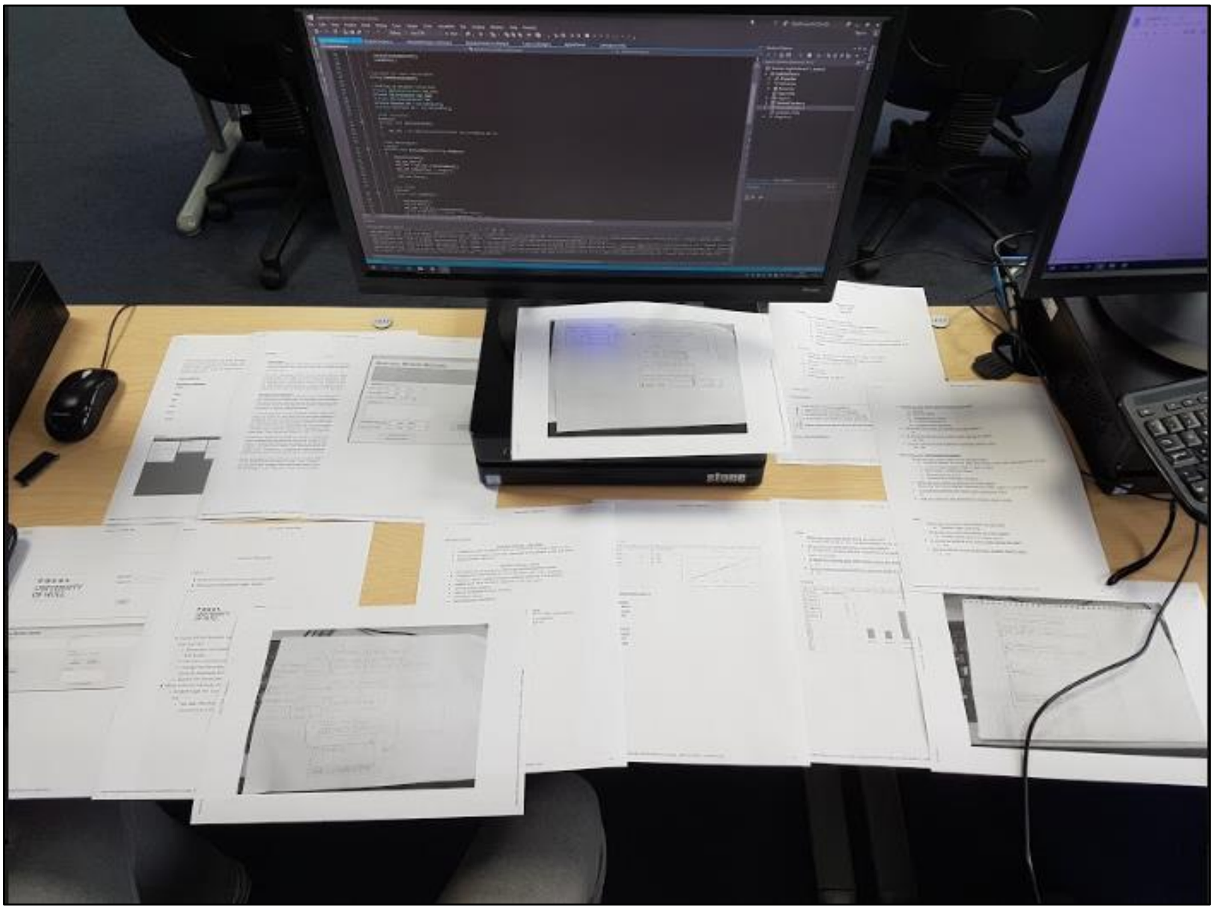


An early Definition of done document was created to outline the general objectives of each of the three broken down levels: sprint (Work Backlog updated.), each user story (User needs satisfied) and also the state of the released product (All acceptance criteria).

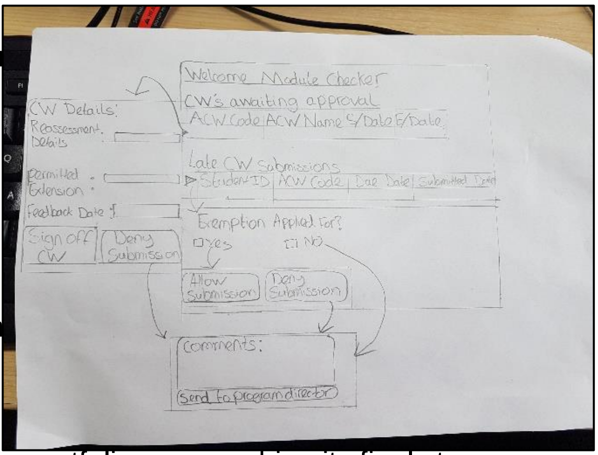
After some initial confusion regarding the definition of done and of the roles of some users, this document was later refined along with the program itself.

As with the previous weeks metrics were captured to view productivity and work backlog and Kanban board updated accordingly.

Snapshot 4



The fourth segment **(Snapshot 4)** saw a drastic redesign of the current system due to a misunderstanding during development that lead to unnecessary features being implemented that would impede the functioning of the essential features of the program. Namely the ability to enter each individual field of the original WPF form through a mimicking form in our program.



Further physical and digital prototyping was do by the development team to accommodate to the new view of the system. Which saw more of the original functionality prioritized over the features that were currently in place in our program.

After receiving feedback our definition of done Document was refined to be more specific to th Project at hand.

At this point work on our script for our tutorial Video had now been completed and work on the portfolio was reaching its final stages.

User stories were revamped to ensure all crucial features were met and then drafted onto our Kanban for the project.

Retrospectives and metrics captured throughout.

The 5th and final segment represents the program and deliverables in their final state as of 03/05/2018. With further changes to the current program which lead to a new interface being created from past prototypes the program was visually ideal but lacked some functionality.

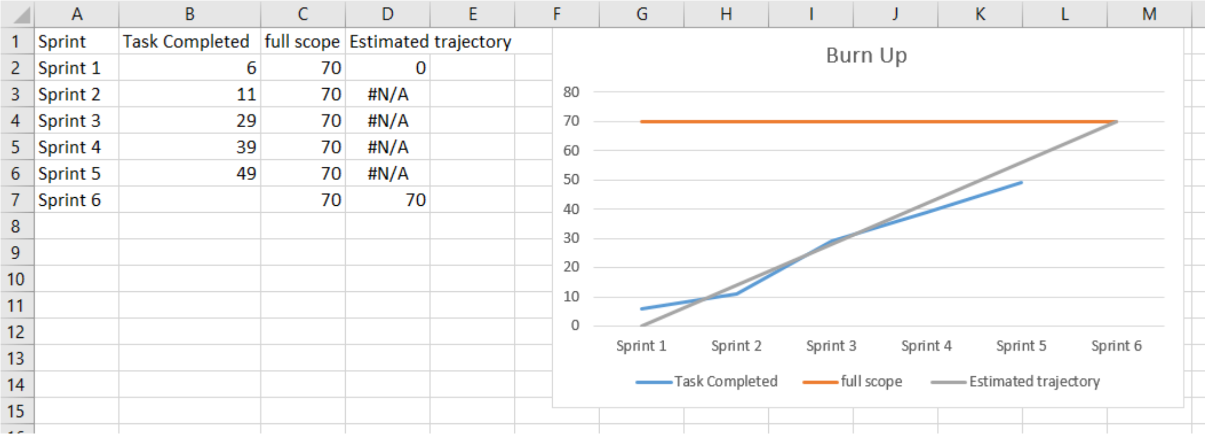
The Portfolio at this stage was edited to include further information regarding each snapshot and use of methodologies and the Video demonstration the program in its current state was captured.

Reflection

Following the initial assignment of our group it was decided that we would make use of **SCRUM** methodology to manage our software development. By having 2-4 20 minute **stand up** meetings per sprint cycle (Mondays progresses check-up, Thursdays progress check-up, Friday mornings snapshot briefing, Friday afternoon snapshot presentations) we were able to efficiently communicate all completed work, tasks that each member plans on tackling next, difficulties found while attempting to complete the task as well as any work members would be placing in another teams way that would affect their progress.

This methodology made the tracking of work completed by which member and when very simple as well as the overall mapping of progression throughout the project. Inclusion of a roadmap time estimation chart helped visualize tasks in the form of days needed for completion. As known within agile, Estimation can only give a rough idea for planning ahead and should not be taken literally due to the high frequency of uncertainty. **Bottom up** techniques where used to break components of the system into separate tasks and uploaded to our Weekly scrum meeting forms to view what needed to be completed for the current sprint.

Other forms of progress tracking that we included in each and every meeting were the current **metrics** for the weeks sprint cycle. In order to track the accumulation of work done over time with the full scope in mind, **burn up charts** were used. In order to track the team's performance for the sprint cycle in relation to previous weeks we made use of **velocity charts**. The metric that both of these charts made use of were task points for which 1 task completed = 1 task point.



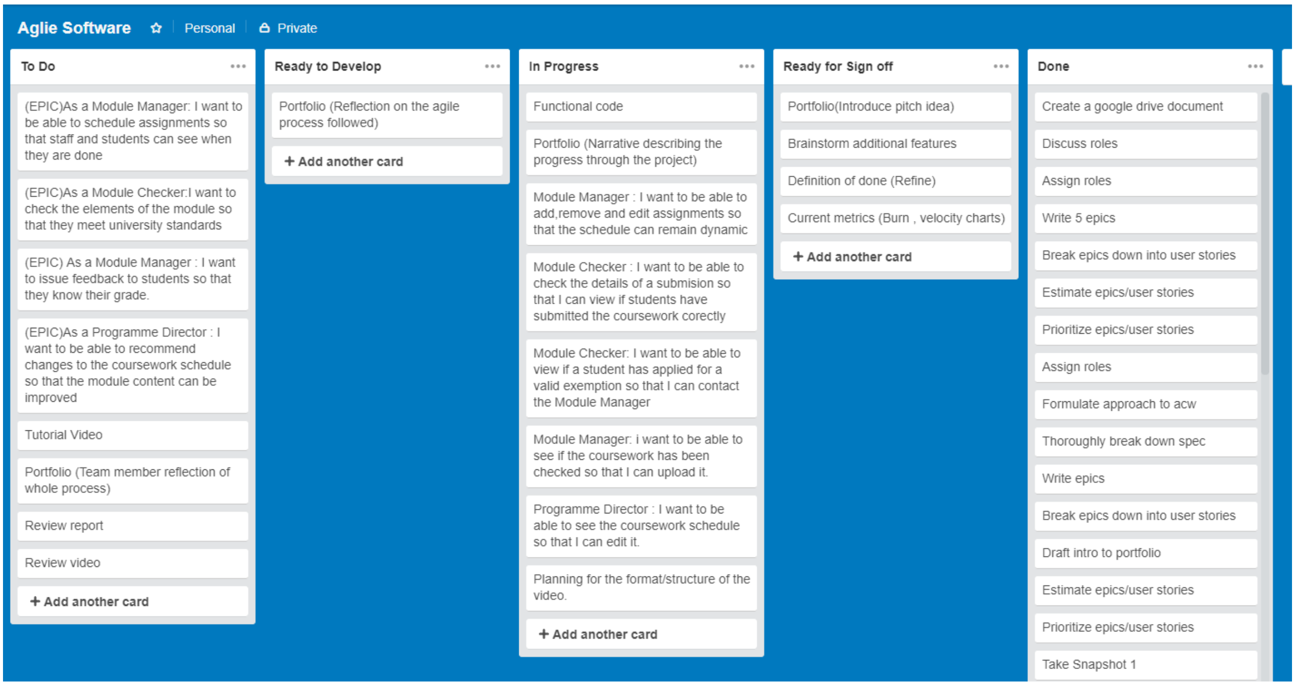
Sprint	Task Completed	full scope	Estimated trajectory
Sprint 1	6	70	0
Sprint 2	11	70	#N/A
Sprint 3	29	70	#N/A
Sprint 4	39	70	#N/A
Sprint 5	49	70	#N/A
Sprint 6		70	70

(Burn up chart tracking accumulation of work done)

The allocation of roles ensured that each member was assigned to a particular team (Development (Product owner), Research, Scrum master, Portfolio) and understood the type of work they would be expected to complete, sometimes independently other times alongside the group as a whole. Members were required to adapt to flexibility and focus ideals in order to remain a vital part of the group and would be asked at times to help with work allocated to another team. This workflow allowed tasks to be completed at a rapid rate. With this we proved that **collaboration** was possible within the group.

Developmental tasks including prototyping and functional coding were left to the development team. Issues lead to this team losing a third of its man power due to unforeseen circumstances but this helped to promote **pair programming** between the remaining members of the development team ensuring quality code was constantly being created.

Our work back log made many transitions while advancing through the project. Initially a physical Kanban containing user stories. This was migrated to google drive so universal access was possible. After this it was maintained on trello where all task management was done and each task was cast into 5 columns being either in column “To Do”, ”Ready To Develop” ,”In Progress”, ”Ready for Sign-Off” or ”Done”. According to its state.



To Do

- (EPIC)As a Module Manager: I want to be able to schedule assignments so that staff and students can see when they are done
- (EPIC)As a Module Checker:I want to check the elements of the module so that they meet university standards
- (EPIC) As a Module Manager : I want to issue feedback to students so that they know their grade.
- (EPIC)As a Programme Director : I want to be able to recommend changes to the coursework schedule so that the module content can be improved
- Tutorial Video
- Portfolio (Team member reflection of whole process)
- Review report
- Review video

Ready to Develop

- Portfolio (Reflection on the agile process followed)

In Progress

- Functional code
- Portfolio (Narrative describing the progress through the project)
- Module Manager : I want to be able to add,remove and edit assignments so that the schedule can remain dynamic
- Module Checker : I want to be able to check the details of a submision so that I can view if students have submitted the coursework corectly
- Module Checker: I want to be able to view if a student has applied for a valid exemption so that I can contact the Module Manager
- Module Manager: i want to be able to see if the coursework has been checked so that I can upload it.
- Programme Director : I want to be able to see the coursework schedule so that I can edit it.
- Planning for the format/structure of the video.

Ready for Sign off

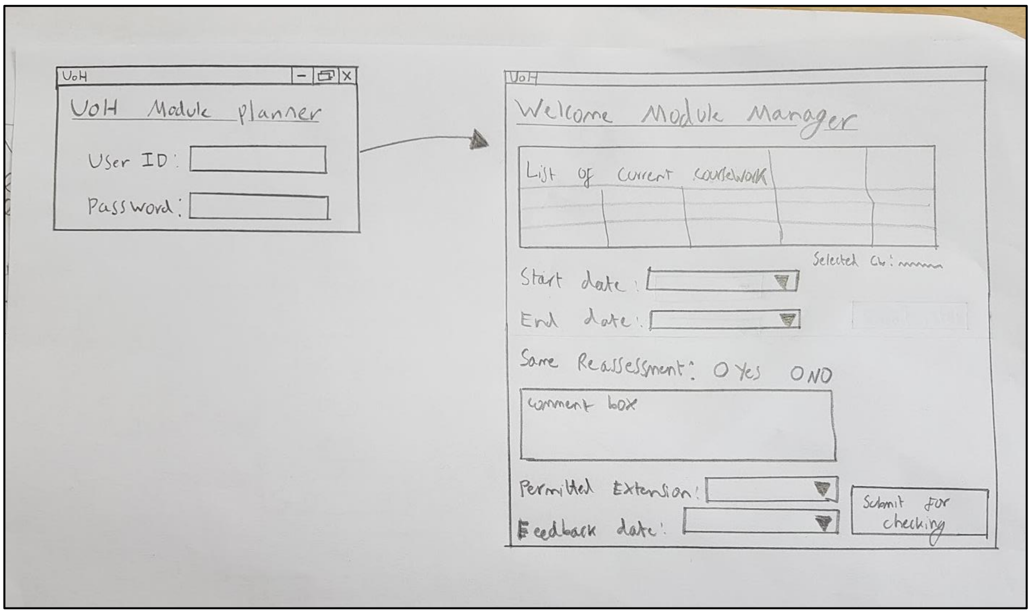
- Portfolio(Introduce pitch idea)
- Brainstorm additional features
- Definition of done (Refine)
- Current metrics (Burn , velocity charts)

Done

- Create a google drive document
- Discuss roles
- Assign roles
- Write 5 epics
- Break epics down into user stories
- Estimate epics/user stories
- Prioritize epics/user stories
- Assign roles
- Formulate approach to acw
- Thoroughly break down spec
- Write epics
- Break epics down into user stories
- Draft intro to portfolio
- Estimate epics/user stories
- Prioritize epics/user stories
- Take Snapshot 1

User research which was completed early in the project lifecycle and further refined as the project drew to a close include the production of many items to enhance the development teams understanding of the project. User stories and epics to delve into basic needs from specific users. **Personas** to identify types of users. Definition of done in order to understand the requirements for completed deliverables (story, sprint, release).

Prototyping of User interfaces and role playing of individual users using the system was done throughout, keeping both the development team and other members of the group aware of current implementations and drawbacks/functionality gained by features.

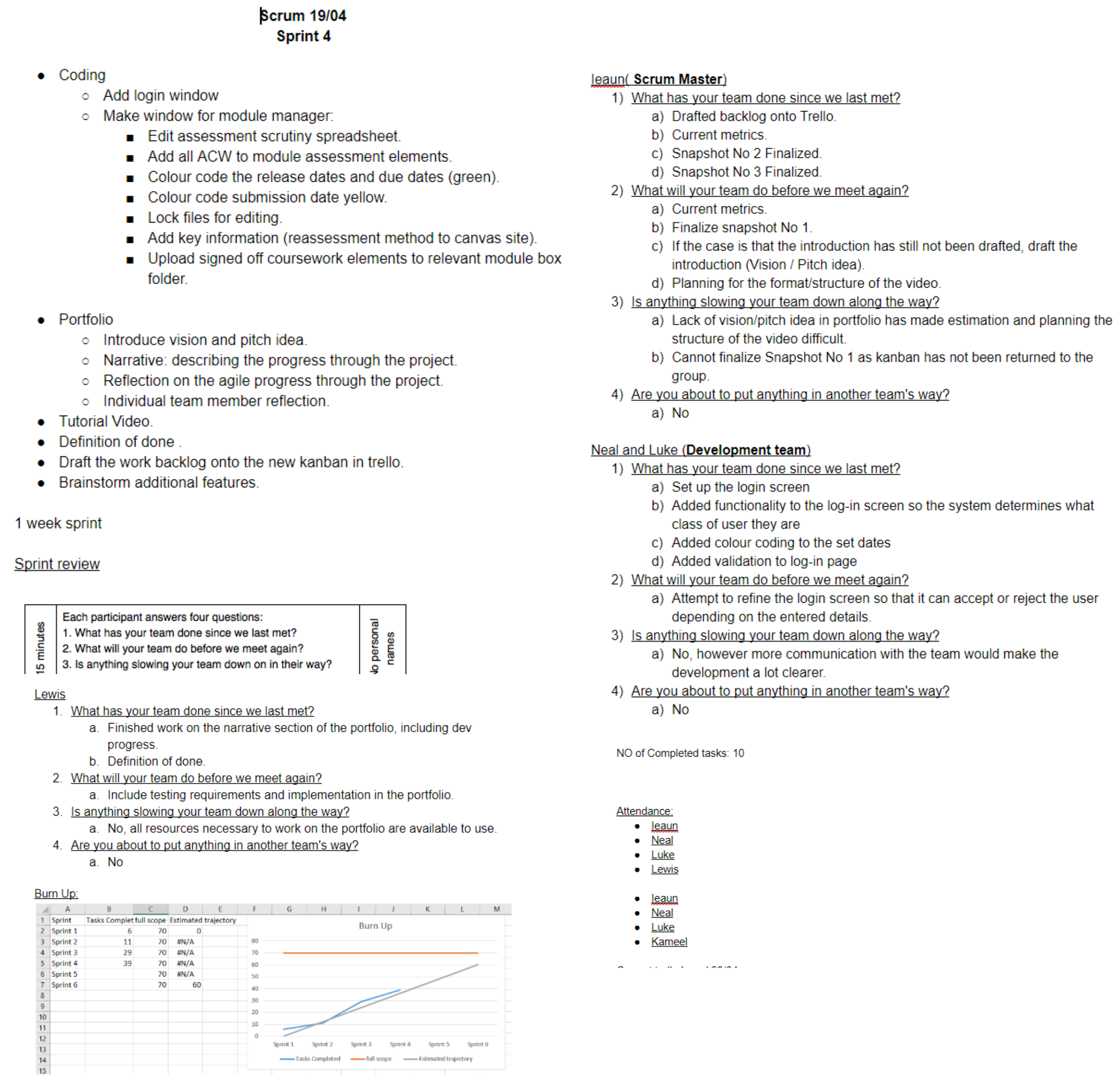


(Early prototypes of the systems interface showing screen flow)

At the halfway checkpoint in our project we were informed of our diversion from the original functionality and adapted our program and what we had thought of as the requirements to ensure the stakeholders would receive a product that met their needs. Key ethos of agile such as ***anticipation of change and adaptation*** were shown here all while ensuring development was **customer centric**. Weekly meetings were held with our customer and questions were constantly being asked by group members to validate the inclusion of features. Further into the project we finalized what features the minimal viable product would have.

As previously mentioned 2-4 meetings were held a week to discuss progress. The information recorded at each of these meetings was then recorded onto sprint forms in the format:

- Date / sprint cycle number
- Backlog of tasks currently in progress or ready to be started.
- Individual retrospectives.
- Metrics.
- Number of completed tasks to be recorded in burn up and velocity charts
- Attendance of meetings.
- Appendix containing Physical evidence of all work completed.



Scrum 19/04
Sprint 4

- Coding
  - Add login window
  - Make window for module manager:
    - Edit assessment scrutiny spreadsheet.
    - Add all ACW to module assessment elements.
    - Colour code the release dates and due dates (green).
    - Colour code submission date yellow.
    - Lock files for editing.
    - Add key information (reassessment method to canvas site).
    - Upload signed off coursework elements to relevant module box folder.

- Portfolio
  - Introduce vision and pitch idea.
  - Narrative: describing the progress through the project.
  - Reflection on the agile progress through the project.
  - Individual team member reflection.
- Tutorial Video.
- Definition of done .
- Draft the work backlog onto the new kanban in trello.
- Brainstorm additional features.

1 week sprint

Sprint review

15 minutes

Each participant answers four questions:
1. What has your team done since we last met?
2. What will your team do before we meet again?
3. Is anything slowing your team down on in their way?

No personal names

Lewis
1. What has your team done since we last met?
   a. Finished work on the narrative section of the portfolio, including dev progress.
   b. Definition of done.
2. What will your team do before we meet again?
   a. Include testing requirements and implementation in the portfolio.
3. Is anything slowing your team down along the way?
   a. No, all resources necessary to work on the portfolio are available to use.
4. Are you about to put anything in another team's way?
   a. No

Burn Up:

	A	B	C	D	E
1	Sprint	Tasks Complet	full scope	Estimated trajectory	
2	Sprint 1	6	70	0	
3	Sprint 2	11	70	#N/A	
4	Sprint 3	29	70	#N/A	
5	Sprint 4	39	70	#N/A	
6	Sprint 5		70	#N/A	
7	Sprint 6		70	60	
8					
9					
10					
11					
12					
13					
14					
15					

Ieaun(**Scrum Master**)
1) What has your team done since we last met?
   a) Drafted backlog onto Trello.
   b) Current metrics.
   c) Snapshot No 2 Finalized.
   d) Snapshot No 3 Finalized.
2) What will your team do before we meet again?
   a) Current metrics.
   b) Finalize snapshot No 1.
   c) If the case is that the introduction has still not been drafted, draft the introduction (Vision / Pitch idea).
   d) Planning for the format/structure of the video.
3) Is anything slowing your team down along the way?
   a) Lack of vision/pitch idea in portfolio has made estimation and planning the structure of the video difficult.
   b) Cannot finalize Snapshot No 1 as kanban has not been returned to the group.
4) Are you about to put anything in another team's way?
   a) No

Neal and Luke (**Development team**)
1) What has your team done since we last met?
   a) Set up the login screen
   b) Added functionality to the log-in screen so the system determines what class of user they are
   c) Added colour coding to the set dates
   d) Added validation to log-in page
2) What will your team do before we meet again?
   a) Attempt to refine the login screen so that it can accept or reject the user depending on the entered details.
3) Is anything slowing your team down along the way?
   a) No, however more communication with the team would make the development a lot clearer.
4) Are you about to put anything in another team's way?
   a) No

NO of Completed tasks: 10

Attendance:
- Ieaun
- Neal
- Luke
- Lewis

- Ieaun
- Neal
- Luke
- Kameel

(Full meeting documentation in appendix)

With all of these factors demonstrated in meetings with our customer it is evident that many agile process principals were followed while completing this project. With each project comes its own unique problems being either related to the product at hand or the team attempting to solve it. The biggest and most costly problems found throughout the project were lack of participation in meetings, lack of attendance in meetings, falsifying completion of work and incompletion of work leading to other members doing as much as 10 times the work of others. This lead to members being spread thin and once this happened quality began to drop.

The program in its current state performs some of the basic functionality and was solved using agile methods that will be embedded into each of the members that helped work on it. Further improvements would need to be made to complete the Software deliverable.

Individual reflection

Lewis:
I believe that despite the issues previously outlined, the fact that every member of the group greatly contributed to the project meant we were largely successful. In addition, our frequent communication with one another face-to-face meant they were almost no issues regarding which work needed to be completed by which date: this was aided even further by the use of a Trello board, as well as access to all documentation via Google Drive.

Ieaun:
Excluding the coding, prototypes and video (Of which was solely completed by Luke and Neal) there is not a single item I did not complete alone or was forced to refine to the point that it would be unrecognizable to the original as lacklustre work was handed in. Most notably:

- Personas
- Estimations
- Acceptance test/criteria
- Metrics each week
- Initial requirements breakdown
- Portfolio (Originally less than 800 words submitted by Lewis as our completed portfolio)
- Definition of done (Kamil handed in the Original which was a copy and paste from the lecture slides.)
- Management of the work backlog and Kanban.
- Retrospectives each week.
- Snapshots 2, 3, 4

Kamil had full access to the canvas group site where had he simply looked he could have seen our meeting times (appendix), All attempts at assigning work to him were met with “I don’t have the correct skills set for that” or 2 week waiting times for 30 words (DOD). Kamil attended 1 meeting outside of lab slots.

Lewis reappeared after a 2 week hiatus. After cancelling on our final meeting and proclaiming that he would upload the portfolio within 2 hours of the completion of said meeting 2 days passed. Finally an 800 word portfolio was submitted on 02/04/2018 which fell on me to refine as it was clear in that document that the writer did not understand the subject matter.
After examination of this portfolio and most items in it I feel It is warranted to say I have a some grasp of agile methodologies to a certain extent and have proved that through my work.

Numesh:
I believe the project was fairly successful as we demonstrated the agile process throughout. Meetings were a huge help with task tracking and making sure we met the deadlines we set, however not all members would attend which made it hard to keep track of work done and some work that was not done was not to standard so had to be redone.

I was working on the development team with Luke which I believe worked well together as we had a fairly even split of the tasks that needed to be coded, there was a slow start as we only had two people working on development. We started off by making some designs for each user story to see that they were fulfilled, but with feedback we realised we had misinterpreted the functions of some of the users and so we developed our designs and improved them to fit the user stories.

Alin was also meant to be a part of the dev team but was not engaged with the coding and did not do the code that was assigned to him so it had to be done by another member.
Pair programming was also very useful as it allowed us to spot changes and mistakes that each other were making allowing us to correct them before they became a major problem. Unfortunately we were not able to complete everything we wanted, possibly due to starting to code later than anticipated or being a member down but we were still able to demonstrate agile practises such as pair planning.

Luke:
The overall process was very successful in terms of the agile process. The weekly scrums were arranged and carried out efficiently and so as a result, each team member was always aware of what work needed to be carried out during each sprint and had a good foresight of how their application would apply to the development of the final deliverables. I would say this was largely down to the effectiveness of the scrum master (Ieaun)'s leadership. In every meeting, he coordinated the sprints excellently and was very good at assessing the previous sprint to plan where we should be at in each stage of the project.

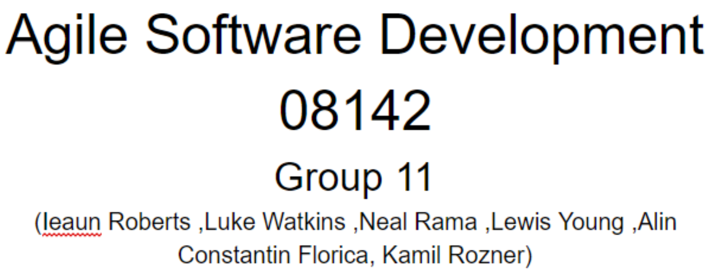
In terms of my designated work, I was tasked to be a part of the development team (Luke, Neal, Alin). In the early weeks of the project, I and Neal each made designs of the code on paper to plan out the design of the forms. We also made early implementations based off the user stories that were drastically altered because of the mistakes we initially made in the retrospective views from every class of user. The code was then gradually developed by myself and Neal until the final hand in, at which point I recorded the tutorial video, a task which we originally designated to Alin. We failed to implement Test driven deign mainly because we did not begin doing this when we first started coding so we thought it would be hard to implement it in an already in-progress project. We did however implement pair programming, which was extremely helpful for quickly developing the project and quickly correcting mistakes. In terms of the overall development, I think it is fair to say Alin contributed nothing.In my opinion every member is responsible for making themselves heard and contributing to the project, something that I believe Alin made no effort to do at any stage of the project.

Alin:
As this project draws to a close I feel that I have learned a lot about applying agile principles in developing software. I believe I’ve gained a better understanding of how regular progress can be achieved each week and what might be expected of me in a real life working environment.
As a team we’ve faced a few communication issues, especially in the beginning, taking us a little while to get everyone together but that usually happens with every group project at university. Once communication was sorted I believe we were able to make steady progress in throughout our regular scrums. I believe overall the project was fairly successful as we managed to accomplish most of the things that we wanted to accomplish.

Kamil:
Due to issues with the timetable, I have joined the development process quite late. At that point, most of the tasks were already assigned to the other team members and in progress.

Appendix

Initial Plan document



Agile Software Development
08142

Group 11

(Ieaun Roberts ,Luke Watkins ,Neal Rama ,Lewis Young ,Alin Constantin Florica, Kamil Rozner)

Introduction

Project Management

Project management describes the planning, initiation and execution of a project. In this section, details on: identifying the project scope, the development process, identifying project risks, the project scheduling information, the work breakdown of the project, and meeting information will all be provided.

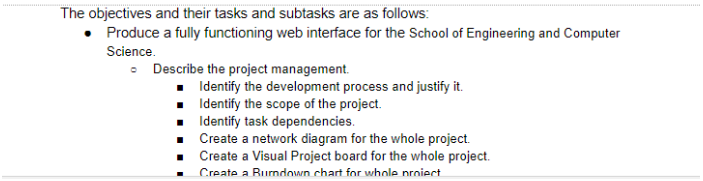
Development Process

Project Goal

The goal of the project is a high level statement which provides the context for what the project is trying to achieve. For this project, the goal is: to design and implement a web based interface for The School of Engineering and Computer Science.

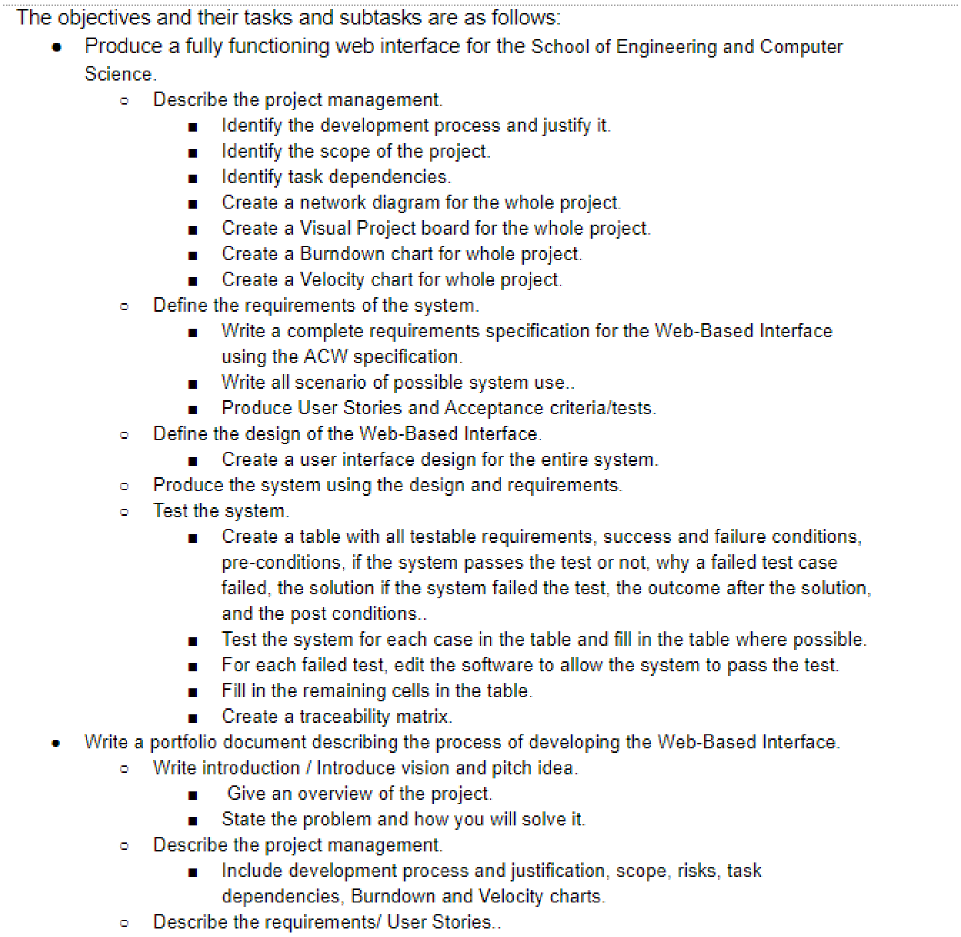
Objectives and Tasks

The project is divided up into multiple objectives, each with their own tasks. An objective describes deliverables that the project will deliver, and is a lower level description than the goal(s) of the project. Each task to complete these objectives provides a lower level description, than the objectives, of how these objectives will be completed. Each task can also be divided into even lower level subtasks, describing how these tasks will be completed.



The objectives and their tasks and subtasks are as follows:

- Produce a fully functioning web interface for the School of Engineering and Computer Science.
  - Describe the project management.
    - Identify the development process and justify it.
    - Identify the scope of the project.
    - Identify task dependencies.
    - Create a network diagram for the whole project.
    - Create a Visual Project board for the whole project.
    - Create a Burndown chart for whole project



The objectives and their tasks and subtasks are as follows:

- Produce a fully functioning web interface for the School of Engineering and Computer Science.
  - Describe the project management.
    - Identify the development process and justify it.
    - Identify the scope of the project.
    - Identify task dependencies.
    - Create a network diagram for the whole project.
    - Create a Visual Project board for the whole project.
    - Create a Burndown chart for whole project.
    - Create a Velocity chart for whole project.
  - Define the requirements of the system.
    - Write a complete requirements specification for the Web-Based Interface using the ACW specification.
    - Write all scenario of possible system use..
    - Produce User Stories and Acceptance criteria/tests.
  - Define the design of the Web-Based Interface.
    - Create a user interface design for the entire system.
  - Produce the system using the design and requirements.
  - Test the system.
    - Create a table with all testable requirements, success and failure conditions, pre-conditions, if the system passes the test or not, why a failed test case failed, the solution if the system failed the test, the outcome after the solution, and the post conditions..
    - Test the system for each case in the table and fill in the table where possible.
    - For each failed test, edit the software to allow the system to pass the test.
    - Fill in the remaining cells in the table.
    - Create a traceability matrix.
- Write a portfolio document describing the process of developing the Web-Based Interface.
  - Write introduction / Introduce vision and pitch idea.
    - Give an overview of the project.
    - State the problem and how you will solve it.
  - Describe the project management.
    - Include development process and justification, scope, risks, task dependencies, Burndown and Velocity charts.
  - Describe the requirements/ User Stories..

Resource Estimation

Roles

Ieaun Roberts - Scrum master
Team - Everyone else
Scrum - using kanban why we are doing this

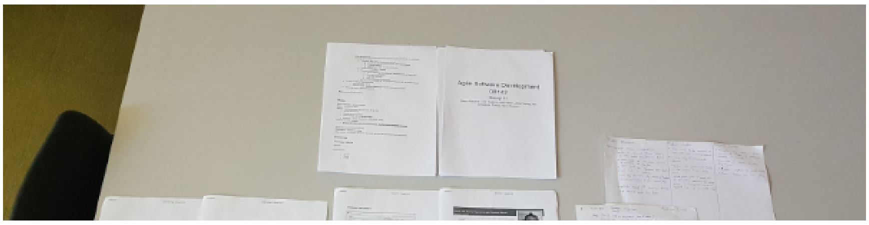
- More structured
- More communication
- Deliverables on a regular basis
- Kanban - more focused on individual tasks

Cons of current system

- ***All go through spec and find some faults/improvements to make***

application/web based solution
Application - easier to make
Web based - probably favoured by users as can be accessed from anywhere

Meetings



Thursday 15/03/18

Sprint 1

Attendance:

- Lewis
- Neal
- Luke
- Ieaun

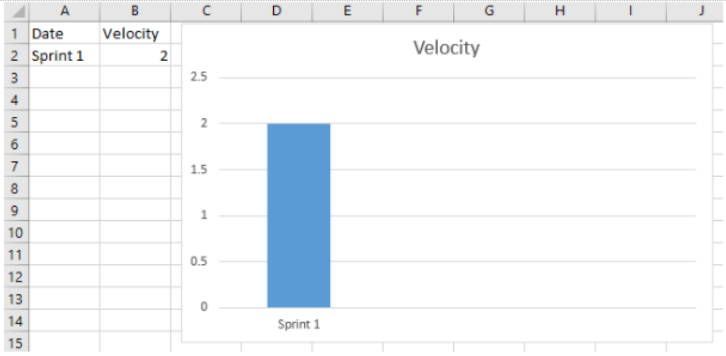
Tasks completed:

- Created a Google Drive with all group members added.
- Wrote up the 5 epics draft from last week and added doc to G-Drive

Tasks for next meeting (16/03/18):

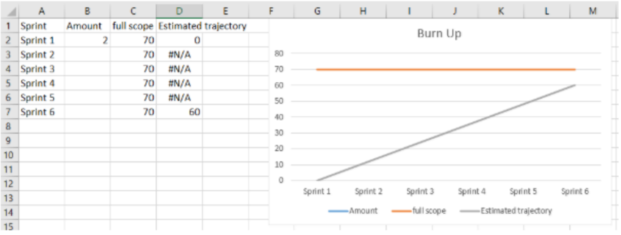
- Review, prioritise and estimate epics/user stories

Velocity



Date	Velocity
Sprint 1	2

Burn UP



Sprint	Amount	full scope	Estimated trajectory
Sprint 1	2	70	0
Sprint 2		70	#N/A
Sprint 3		70	#N/A
Sprint 4		70	#N/A
Sprint 5		70	#N/A
Sprint 6		70	60

Friday 16/3/18

Sprint 1

Attendance:

- Lewis
- Luke
- Neal

Tasks completed:

- Assigned Scrum Master
- Informally discussed role allocation
- Read spec
- Decided on approach to task

Tasks for next meeting (22/03/18):

- Allocate roles
- Thoroughly break down spec
- List epics/user stories
- Draft intro to portfolio
- Add cons to current system in G-Drive

Notes:

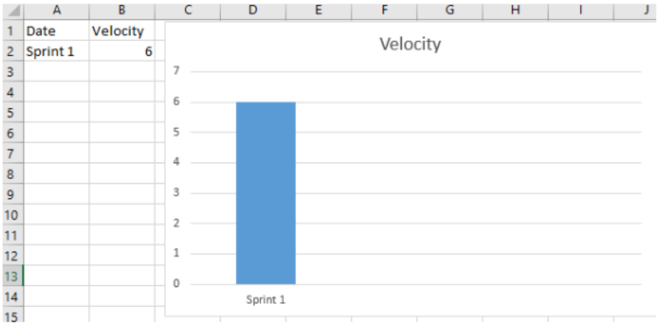
- Next meeting will take place on Thursday 22nd March 1-2, in GLR1, 3rd floor

- Draft intro to portfolio
- Add cons to current system in G-Drive

Notes:

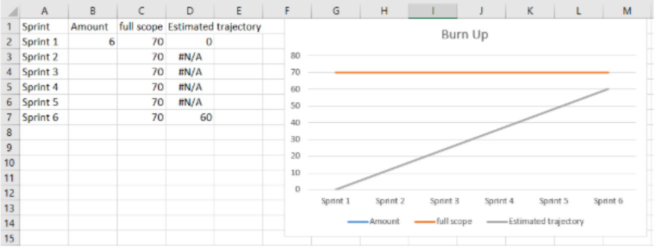
- Next meeting will take place on Thursday 22nd March 1-2, in GLR1, 3rd floor library.

Velocity



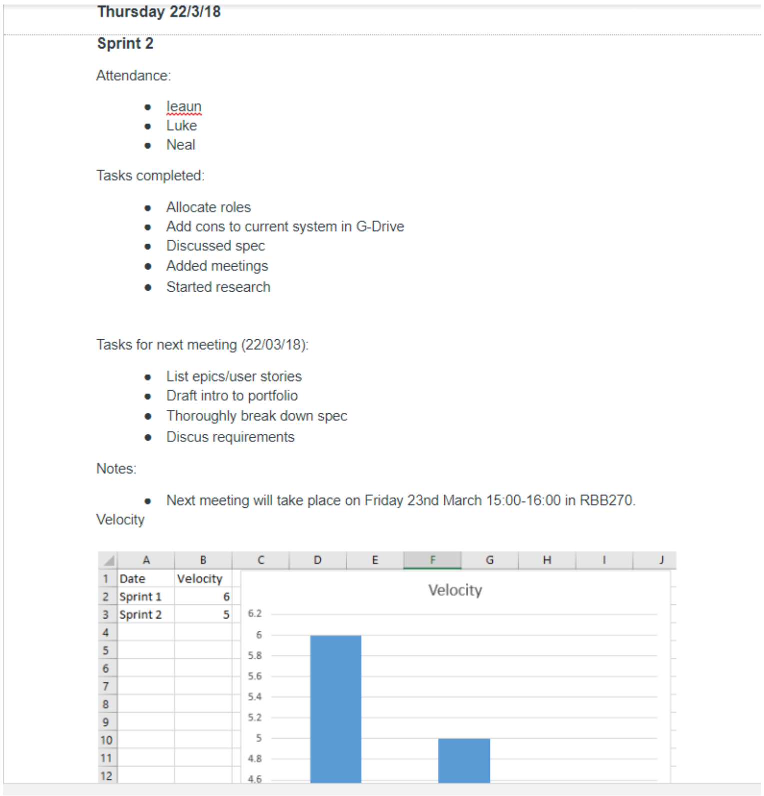
Date	Velocity
Sprint 1	6

Burn Up



Sprint	Amount	full scope	Estimated trajectory
Sprint 1	6	70	0
Sprint 2		70	#N/A
Sprint 3		70	#N/A
Sprint 4		70	#N/A
Sprint 5		70	#N/A
Sprint 6		70	60

Thursday 22/3/18



Thursday 22/3/18

Sprint 2

Attendance:

- leaun
- Luke
- Neal

Tasks completed:

- Allocate roles
- Add cons to current system in G-Drive
- Discussed spec
- Added meetings
- Started research

Tasks for next meeting (22/03/18):

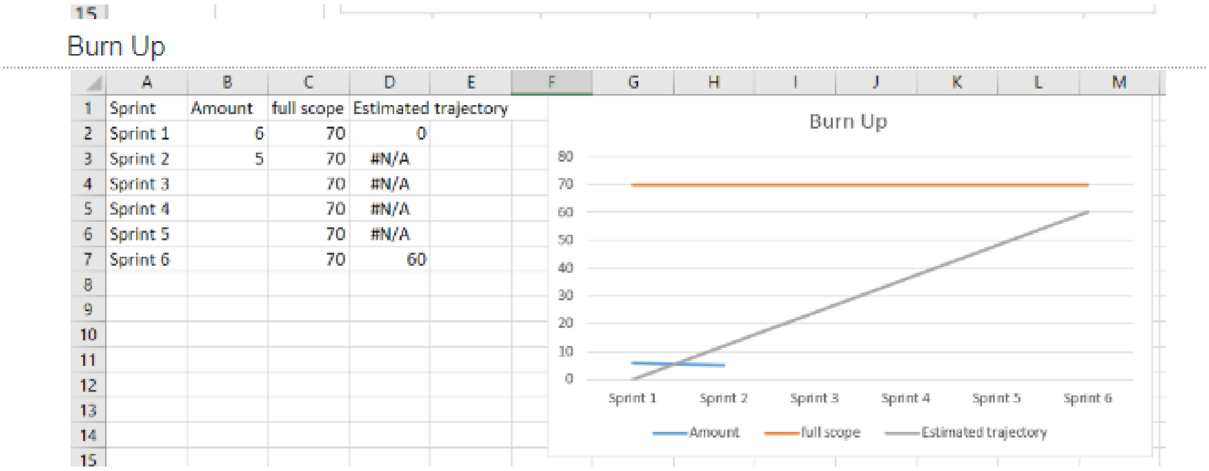
- List epics/user stories
- Draft intro to portfolio
- Thoroughly break down spec
- Discus requirements

Notes:

- Next meeting will take place on Friday 23nd March 15:00-16:00 in RBB270.

Velocity

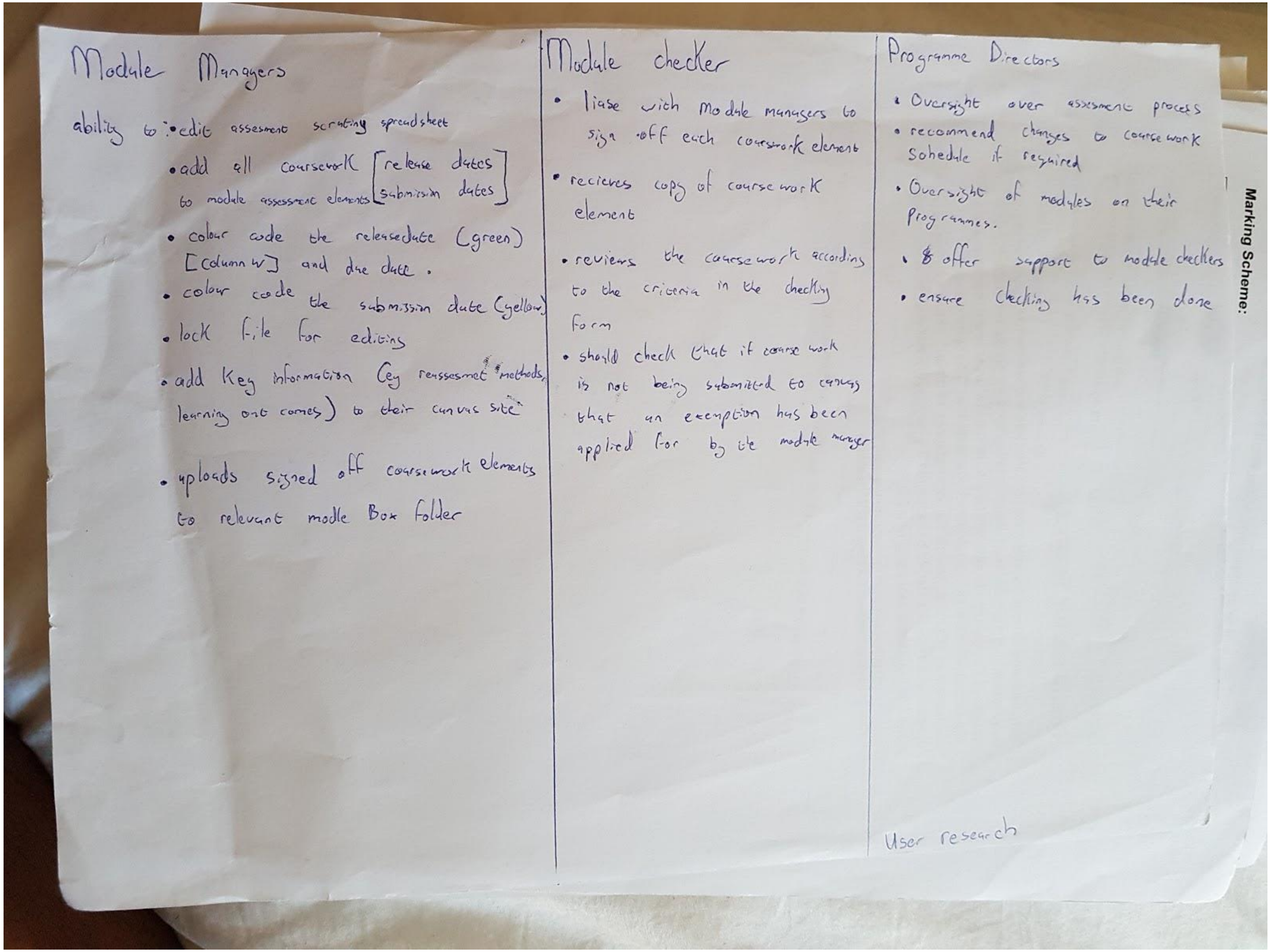
Date	Velocity
Sprint 1	6
Sprint 2	5



Burn Up

Sprint	Amount	full scope	Estimated trajectory
Sprint 1	6	70	0
Sprint 2	5	70	#N/A
Sprint 3		70	#N/A
Sprint 4		70	#N/A
Sprint 5		70	#N/A
Sprint 6		70	60

Requirement breakdown



Module Managers

ability to:
- edit assesment scruting spreadsheet
- add all coursework [release dates / submission dates] to module assessment elements
- colour code the releasedate (green) [column W] and due date.
- colour code the submission date (yellow)
- lock file for editing
- add Key information (ey reassesmet methods, learning out comes) to their canvas site
- uploads signed off coursework elements to relevant modle Box folder

Module checker

- liase with module managers to sign off each coursework element
- recieves copy of coursework element
- reviews the coursework according to the criteria in the checking form
- should check that if course work is not being submitted to canvas that an exemption has been applied for by the module manager

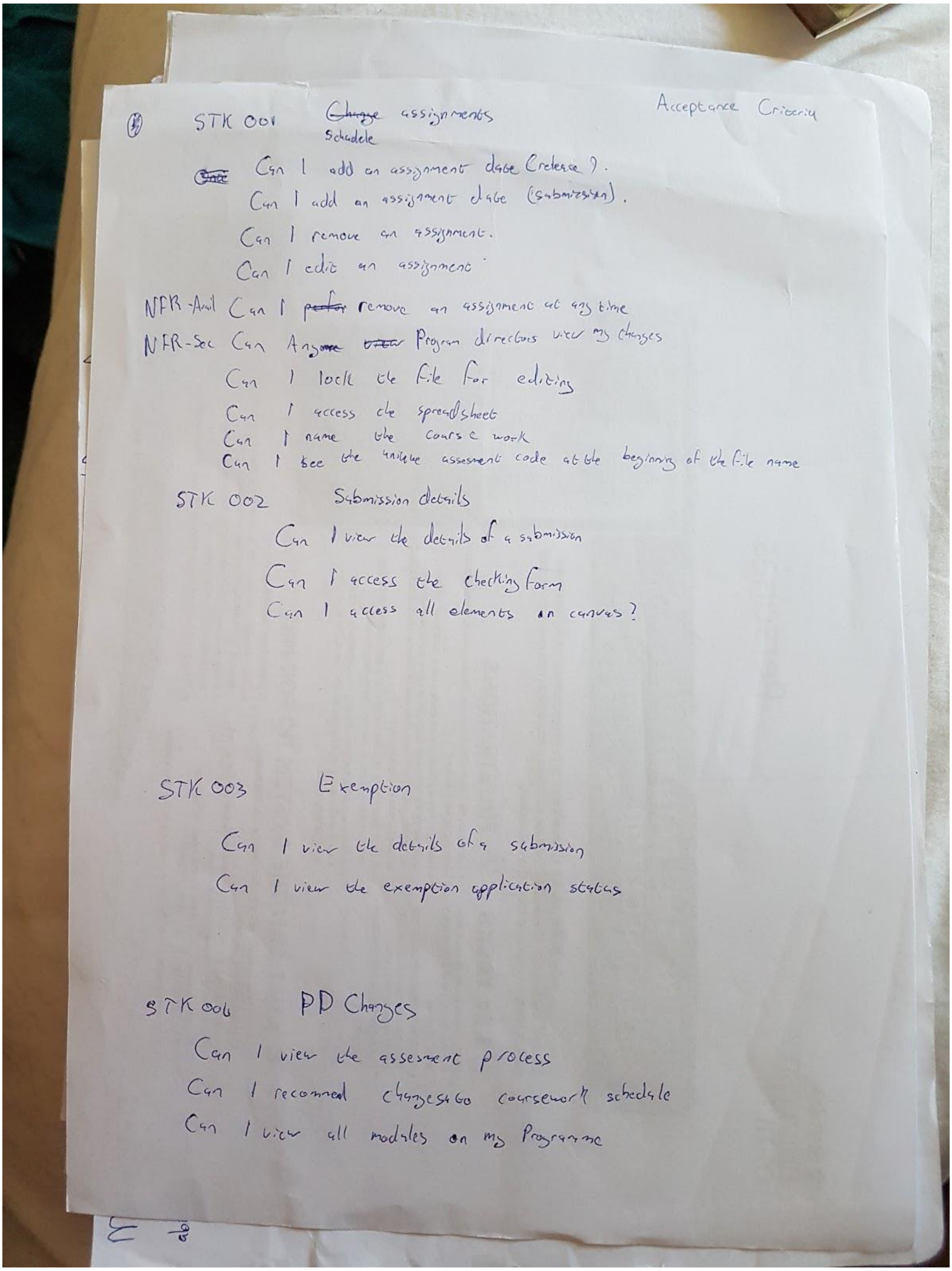
Programme Directors

- Oversight over assesment process
- recommend changes to course work schedule if required
- Oversight of modules on their Programmes.
- $ offer support to module checkers
- ensure checking has been done

User research

Marking Scheme:

Acceptance tests/ criteria



Acceptance Criteria

STK 001 ~~Change~~ assignments
Schadele

~~Date~~ Can I add an assignment date (release ?.

Can I add an assignment date (submission).

Can I remove an assignment.

Can I edit an assignment

NFR-Avail Can I ~~perfor~~ remove an assignment at any time

NFR-Sec Can Any~~one~~ ~~view~~ Program directors view my changes

Can I lock the file for editing

Can I access the spreadsheet

Can I name the course work

Can I see the unique assesment code at the beginning of the file name

STK 002 Submission details

Can I view the details of a submission

Can I access the checking form

Can I access all elements on canvas?

STK 003 Exemption

Can I view the details of a submission

Can I view the exemption application status

STK 006 PD Changes

Can I view the assesment process

Can I recommend changes to coursework schedule

Can I view all modules on my Programme

Personas

Benedict, the School of Engineering and Computer Science Module Manager

"My main focus while working here is on assignment scheduling and the issuing of feedback"

Module managers like Benedict are needed to schedule assignments and issue feedback

Key Tasks

- Add all coursework release/submission dates.
- Colour code mentioned dates in spreadsheet
- Lock files for editing
- Add key information regarding items such as reassignment methods or learning outcomes to their canvas site.
- Upload signed off course work elements to relevant module box folder

Frustrations

- Has communicated that current workflow plan needs improvement.
- Inability to distinguish release/ submission dates at a first glance.
- File can currently be edited by multiple staff members at once causing overwriting of data.
- Lack of centralized data repository.
- Inefficient method of sending copy of course work elements to checkers.

What can the system do for me?

1. Allow user to seamlessly edit, remove or add any necessary fields.
2. A colour coded system can clear some of the confusion that exists.
3. Ensure communication between Module Managers and Module Checkers.

Martin, the School of Engineering and Computer Science Module Checker

"Once you add me as a course teacher on canvas your work will be as good as checked"

Module Checkers like Martin are needed to liaise with Module Managers to sign off and check elements of the module.

Key Tasks

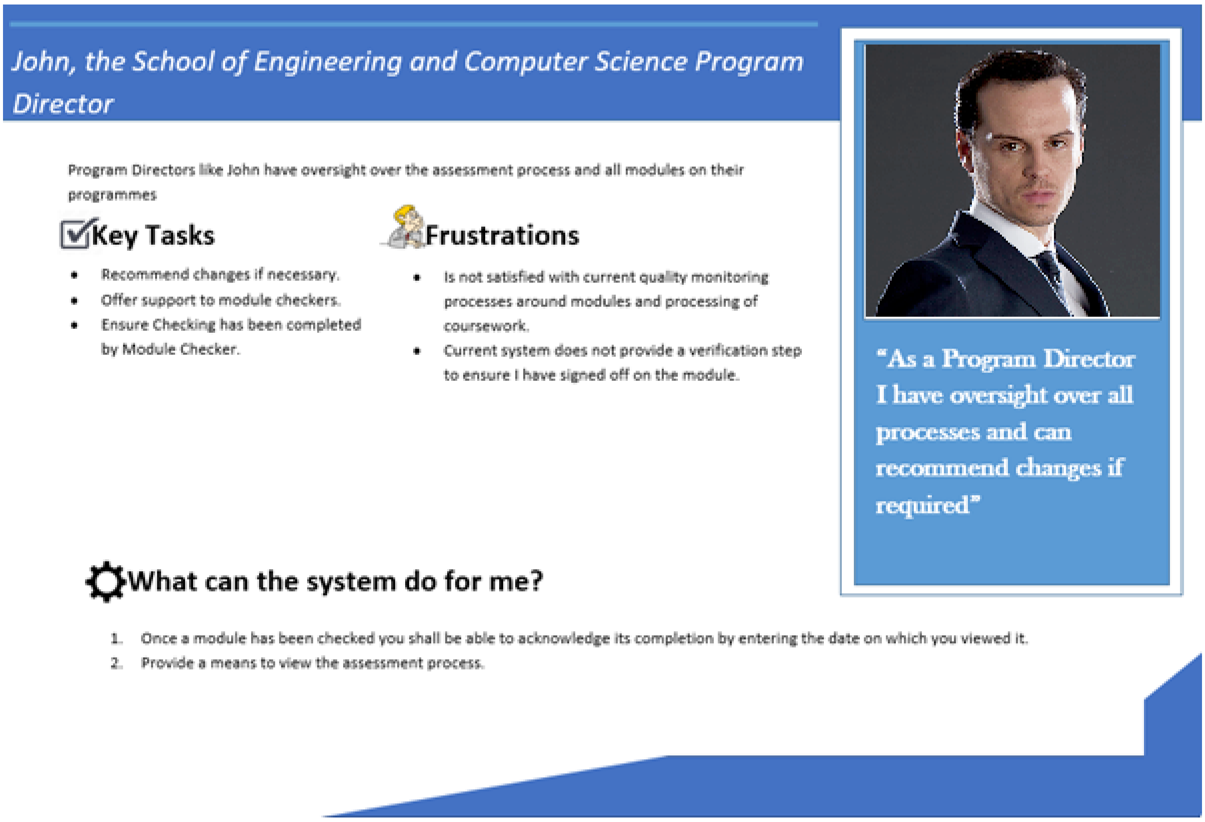
- Reviews coursework according to the criteria in the checking form.
- Checking if exemptions have been applied for if course work is not submitted.
- Contacts John and Pippa regarding out of the ordinary submissions.

Frustrations

- Having trouble understanding current forms without help from module managers.

What can the system do for me?

1. Provide a means to easily view whether or not an application for an exemption has been applied for.
2. Provide an efficient form to view current coursework.



John, the School of Engineering and Computer Science Program Director

Program Directors like John have oversight over the assessment process and all modules on their programmes

☑Key Tasks

- Recommend changes if necessary.
- Offer support to module checkers.
- Ensure Checking has been completed by Module Checker.

Frustrations

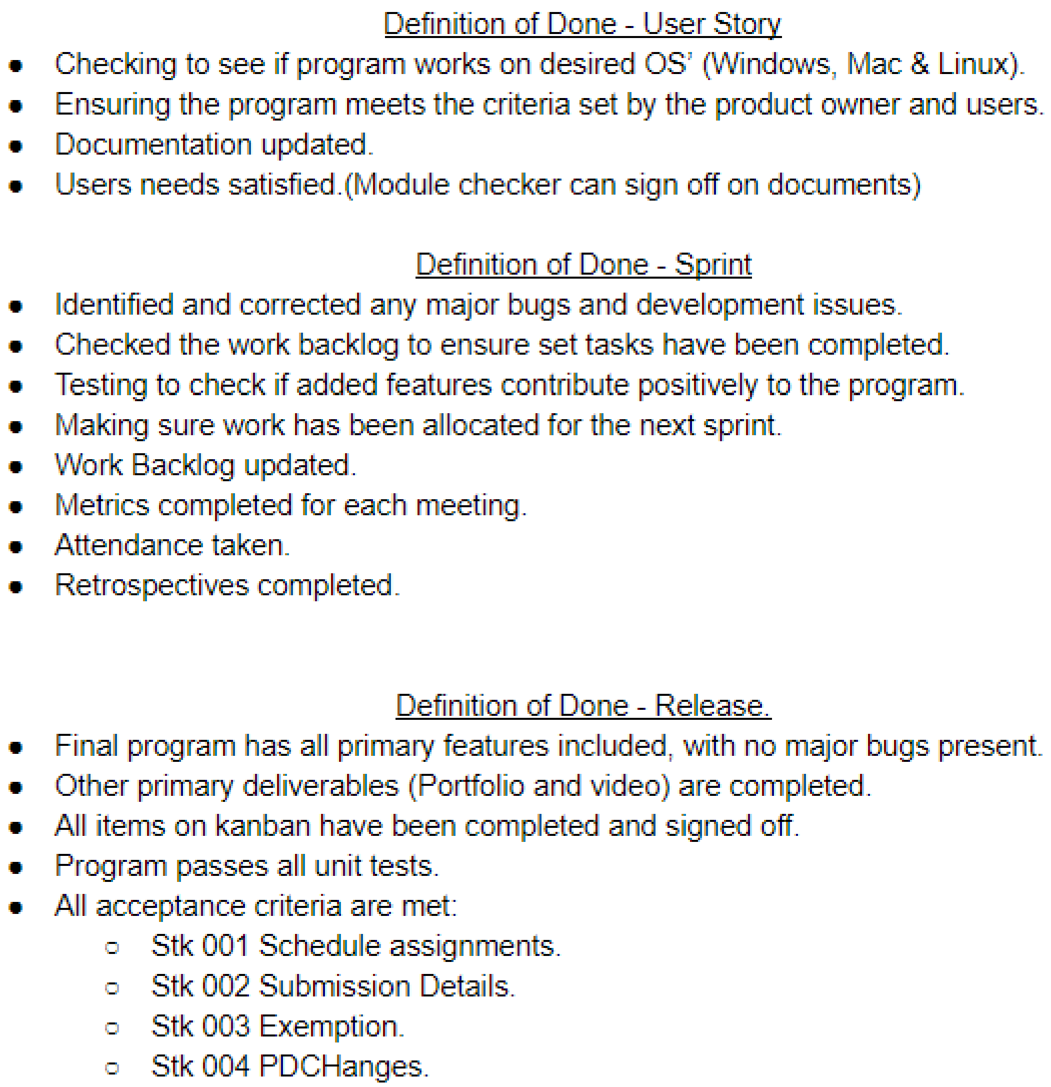
- Is not satisfied with current quality monitoring processes around modules and processing of coursework.
- Current system does not provide a verification step to ensure I have signed off on the module.

> "As a Program Director I have oversight over all processes and can recommend changes if required"

What can the system do for me?

1. Once a module has been checked you shall be able to acknowledge its completion by entering the date on which you viewed it.
2. Provide a means to view the assessment process.

Definition of done



Definition of Done - User Story

- Checking to see if program works on desired OS’ (Windows, Mac & Linux).
- Ensuring the program meets the criteria set by the product owner and users.
- Documentation updated.
- Users needs satisfied.(Module checker can sign off on documents)

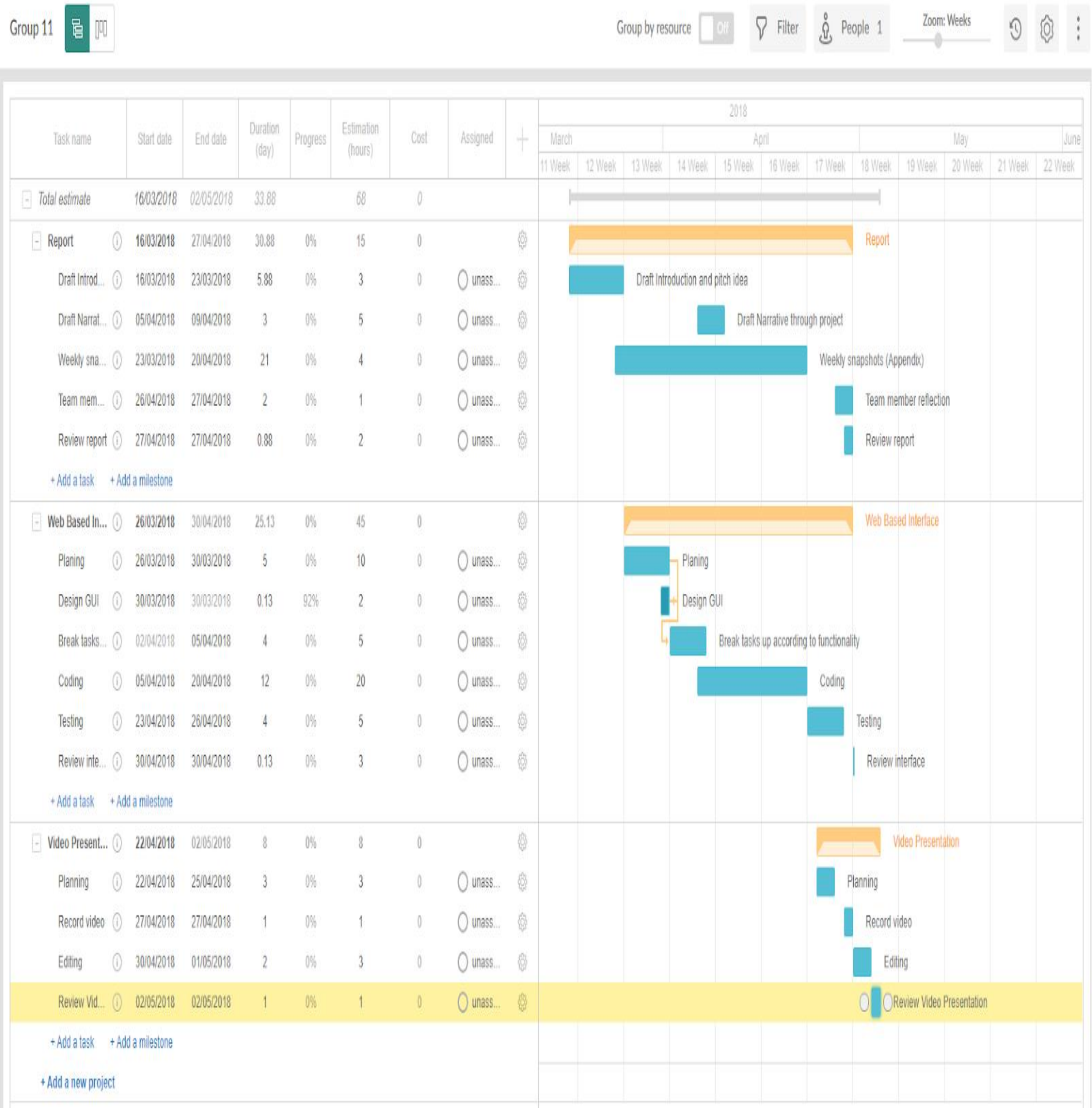
Definition of Done - Sprint

- Identified and corrected any major bugs and development issues.
- Checked the work backlog to ensure set tasks have been completed.
- Testing to check if added features contribute positively to the program.
- Making sure work has been allocated for the next sprint.
- Work Backlog updated.
- Metrics completed for each meeting.
- Attendance taken.
- Retrospectives completed.

Definition of Done - Release.

- Final program has all primary features included, with no major bugs present.
- Other primary deliverables (Portfolio and video) are completed.
- All items on kanban have been completed and signed off.
- Program passes all unit tests.
- All acceptance criteria are met:
    - Stk 001 Schedule assignments.
    - Stk 002 Submission Details.
    - Stk 003 Exemption.
    - Stk 004 PDCHanges.

Estimated time planning

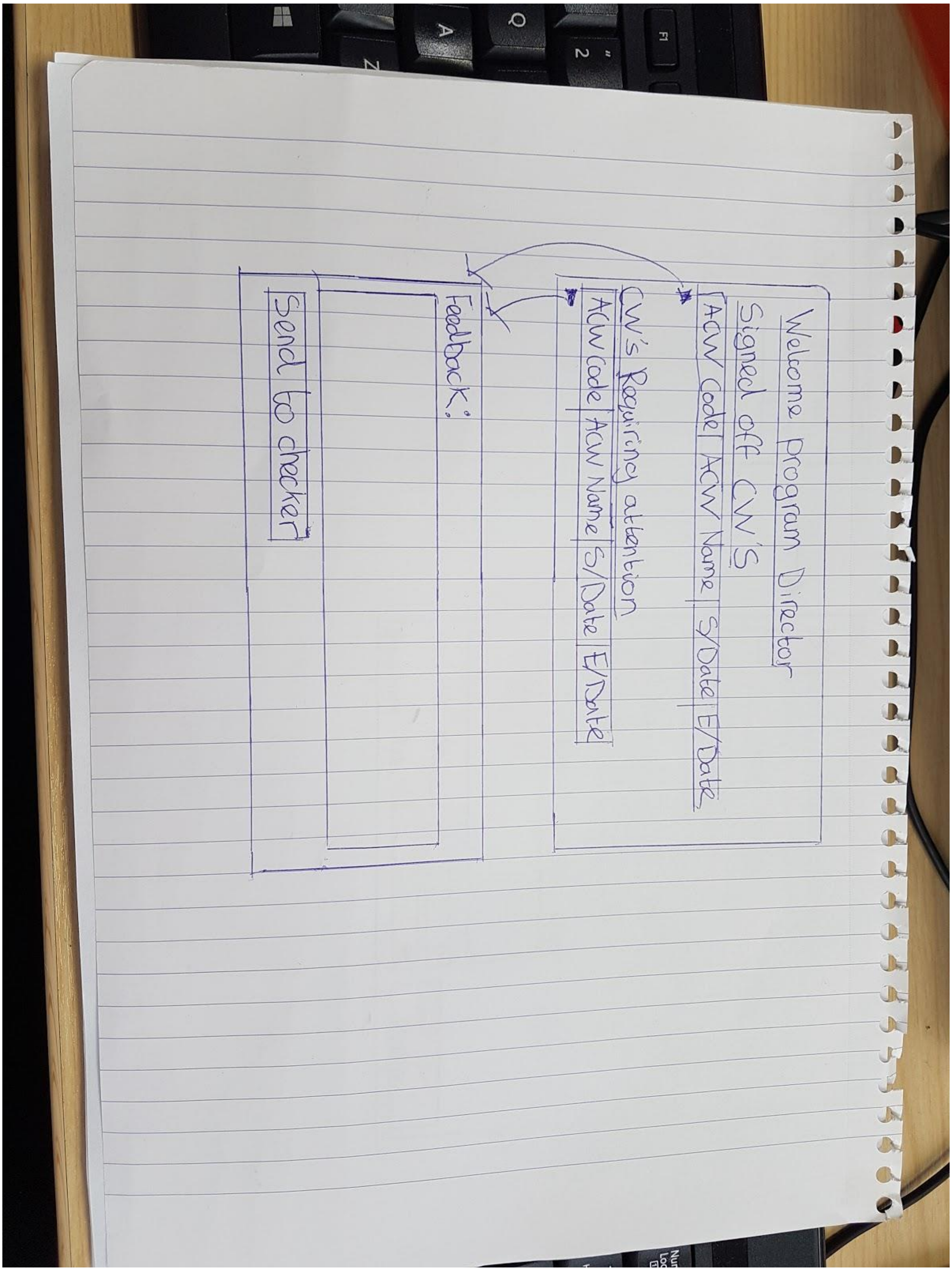


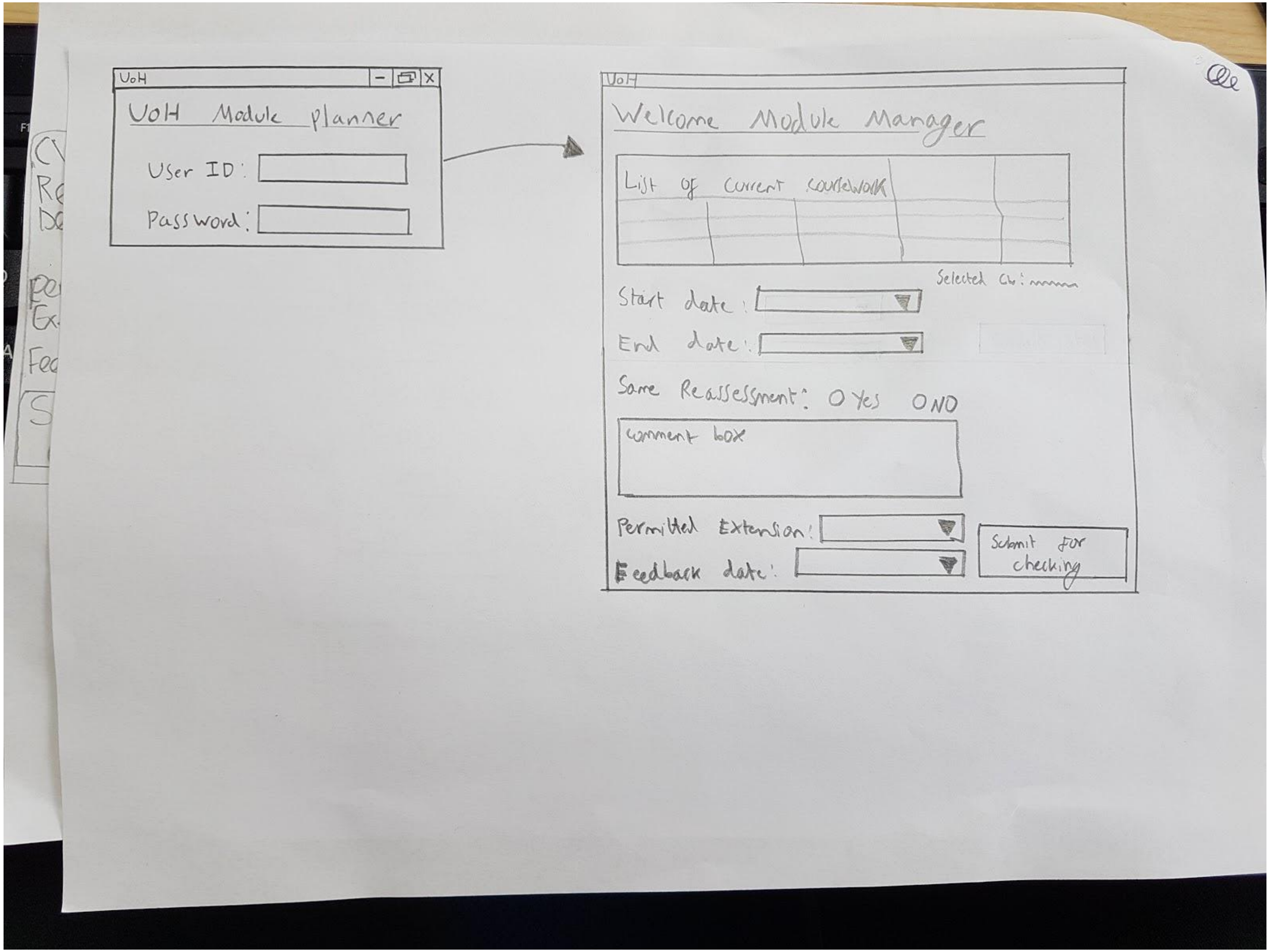
Task name	Start date	End date	Duration (day)	Progress	Estimation (hours)	Cost	Assigned
Total estimate	*16/03/2018*	*02/05/2018*	*33.88*		*68*	*0*	
Report	16/03/2018	27/04/2018	30.88	0%	15	0	
Draft Introd...	16/03/2018	23/03/2018	5.88	0%	3	0	○ unass...
Draft Narrat...	05/04/2018	09/04/2018	3	0%	5	0	○ unass...
Weekly sna...	23/03/2018	20/04/2018	21	0%	4	0	○ unass...
Team mem...	26/04/2018	27/04/2018	2	0%	1	0	○ unass...
Review report	27/04/2018	27/04/2018	0.88	0%	2	0	○ unass...
+ Add a task + Add a milestone							
Web Based In...	26/03/2018	30/04/2018	25.13	0%	45	0	
Planing	26/03/2018	30/03/2018	5	0%	10	0	○ unass...
Design GUI	30/03/2018	30/03/2018	0.13	92%	2	0	○ unass...
Break tasks...	02/04/2018	05/04/2018	4	0%	5	0	○ unass...
Coding	05/04/2018	20/04/2018	12	0%	20	0	○ unass...
Testing	23/04/2018	26/04/2018	4	0%	5	0	○ unass...
Review inte...	30/04/2018	30/04/2018	0.13	0%	3	0	○ unass...
+ Add a task + Add a milestone							
Video Present...	22/04/2018	02/05/2018	8	0%	8	0	
Planning	22/04/2018	25/04/2018	3	0%	3	0	○ unass...
Record video	27/04/2018	27/04/2018	1	0%	1	0	○ unass...
Editing	30/04/2018	01/05/2018	2	0%	3	0	○ unass...
Review Vid...	02/05/2018	02/05/2018	1	0%	1	0	○ unass...
+ Add a task + Add a milestone							
+ Add a new project							

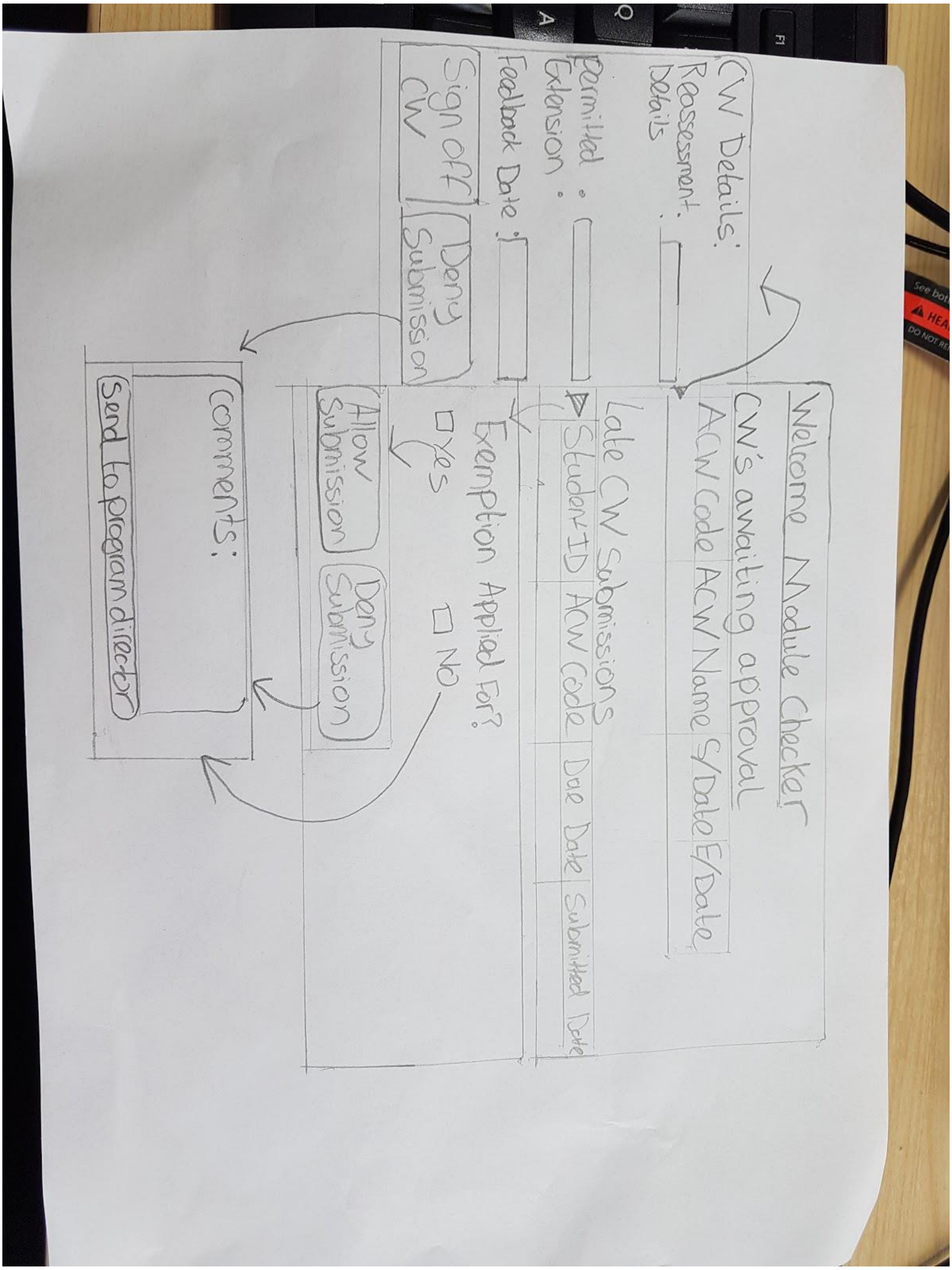
Prioritized and estimated stories, epics and themes.

1. STK 001 - Schedule assignments - 6 days
2. STK 002 - Submission details - 3 days
3. STK 003 - Exemption - 2 days
4. STK 004 - PD Changes -2 days

Prototypes


Welcome Program Director
Signed off CW'S
ACW Code | ACW Name | S/Date | E/Date
CW's Requiring attention
ACW Code | ACW Name | S/Date | E/Date
Feedback:
Send to checker


UoH
UoH Module planner
User ID:
Password:
UoH
Welcome Module Manager
List of current coursework
Selected cw: ~~~~
Start date:
End date:
Same Reassessment: ○ Yes ○ NO
comment box
Permitted Extension:
Feedback date:
Submit for checking

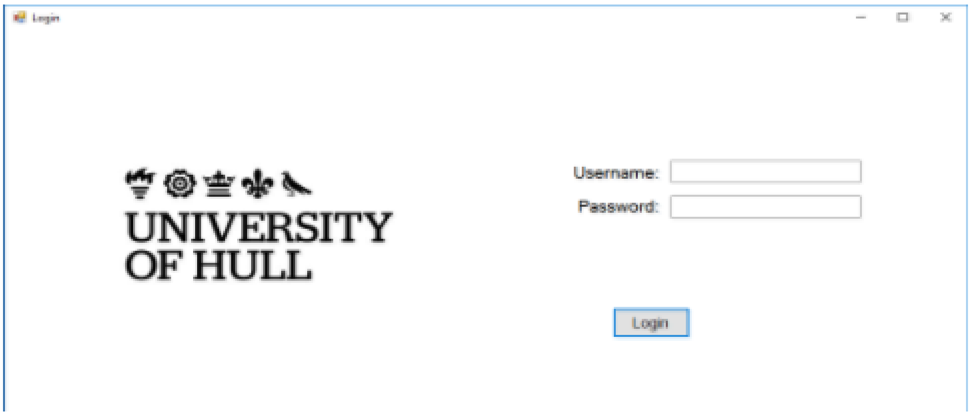

Welcome Module Checker
CW's awaiting approval
ACW Code
ACW Name
S/Date
E/Date
Late CW Submissions
Student ID
ACW Code
Due Date
Submitted Date
Exemption Applied for?
□ Yes
□ No
Allow Submission
Deny Submission
CW Details:
Reassessment Details :
Permitted Extension :
Feedback Date :
Sign off CW
Deny Submission
Comments:
Send to program director

Video Plan

Tutorial Video plan

Layout:

- Brief introduction of the overall system
- Showcase and explain login screen



- Showcase the Module manager UI and walkthrough the user journey
  - Showcase coursework selection using the Start and End Dates
  - Use the comments box to submit a comment
  - Change the Permitted Extension date and Feedback Date to showcase functionality
  - Submit the showcased coursework for checking
- Move onto the Module checker user journey
  - Walkthrough the Courseworks awaiting approval list
    - Go over the three details boxes once a coursework is selected

Retrospective meeting

Scrum 19/04
Sprint 4

- Coding
  - Add login window
  - Make window for module manager:
    - Edit assessment scrutiny spreadsheet.
    - Add all ACW to module assessment elements.
    - Colour code the release dates and due dates (green).
    - Colour code submission date yellow.
    - Lock files for editing.
    - Add key information (reassessment method to canvas site).
    - Upload signed off coursework elements to relevant module box folder.

- Portfolio
  - Introduce vision and pitch idea.
  - Narrative: describing the progress through the project.
  - Reflection on the agile progress through the project.
  - Individual team member reflection.
- Tutorial Video.
- Definition of done .
- Draft the work backlog onto the new kanban in trello.
- Brainstorm additional features.

1 week sprint

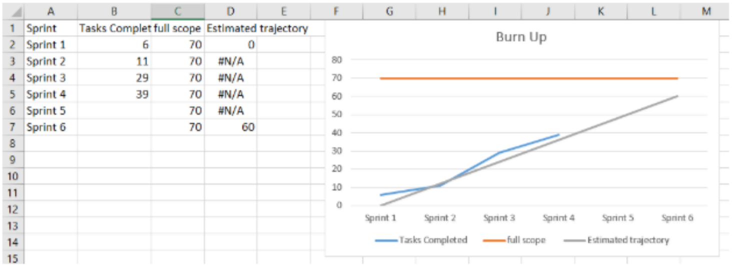
Sprint review

15 minutes	Each participant answers four questions: 1. What has your team done since we last met? 2. What will your team do before we meet again? 3. Is anything slowing your team down on in their way? 4. Are you about to put anything in another team's way?	No personal names
As needed	Resolve problems and discuss issues on the team backlog.	

Lewis

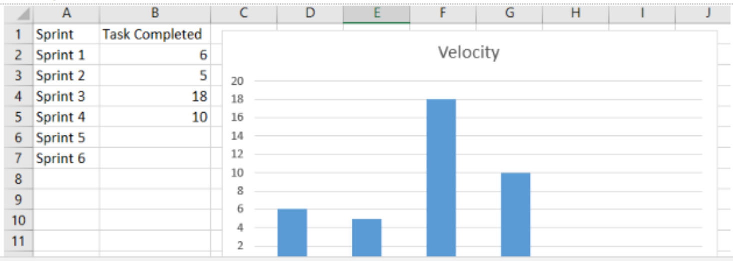
1. What has your team done since we last met?
   a. Finished work on the narrative section of the portfolio, including dev progress.
   b. Definition of done.
2. What will your team do before we meet again?
   a. Include testing requirements and implementation in the portfolio.
3. Is anything slowing your team down along the way?
   a. No, all resources necessary to work on the portfolio are available to use.
4. Are you about to put anything in another team's way?
   a. No

Burn Up:



Sprint	Tasks Complet	full scope	Estimated trajectory
Sprint 1	6	70	0
Sprint 2	11	70	#N/A
Sprint 3	29	70	#N/A
Sprint 4	39	70	#N/A
Sprint 5		70	#N/A
Sprint 6		70	60

Velocity:



Sprint	Task Completed
Sprint 1	6
Sprint 2	5
Sprint 3	18
Sprint 4	10
Sprint 5	
Sprint 6	

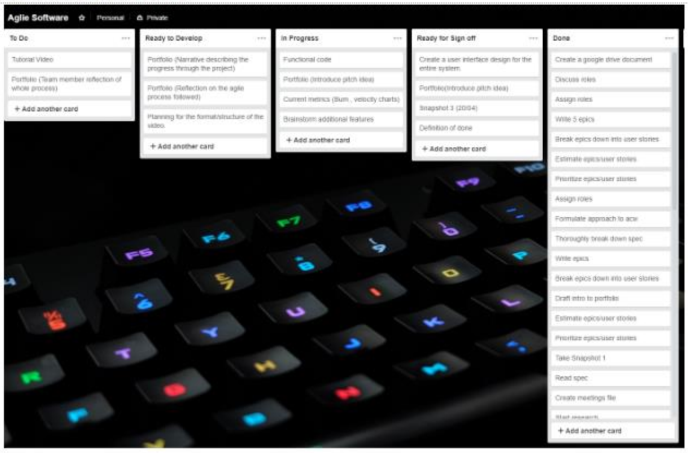
NO of Completed tasks: 10

Attendance:

- Ieaun
- Neal
- Luke
- Lewis

- Ieaun
- Neal
- Luke
- Kameel

Current trello board 20/04



Agile Software · Personal · Private

To Do
- Tutorial Video
- Portfolio (Team member reflection of whole process)
- + Add another card

Ready to Develop
- Portfolio (Narrative describing the progress through the project)
- Portfolio (Reflection on the agile process followed)
- Planning for the format/structure of the video.
- + Add another card

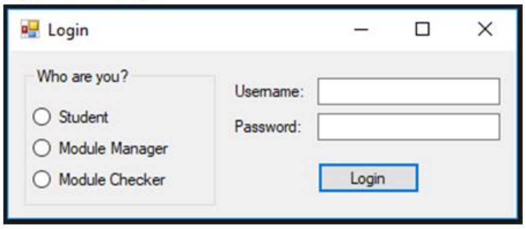
In Progress
- Functional code
- Portfolio (introduce pitch idea)
- Current metrics (Burn , velocity charts)
- Brainstorm additional features
- + Add another card

Ready for Sign off
- Create a user interface design for the entire system.
- Portfolio(introduce pitch idea)
- Snapshot 3 (20/04)
- Definition of done
- + Add another card

Done
- Create a google drive document
- Discuss roles
- Assign roles
- Write 5 epics
- Break epics down into user stories
- Estimate epics/user stories
- Prioritize epics/user stories
- Assign roles
- Formulate approach to acw
- Thoroughly break down spec
- Write epics
- Break epics down into user stories
- Draft intro to portfolio
- Estimate epics/user stories
- Prioritize epics/user stories
- Take Snapshot 1
- Read spec
- Create meetings file
- Start research
- + Add another card

Login screen for program

Login screen for program



Login

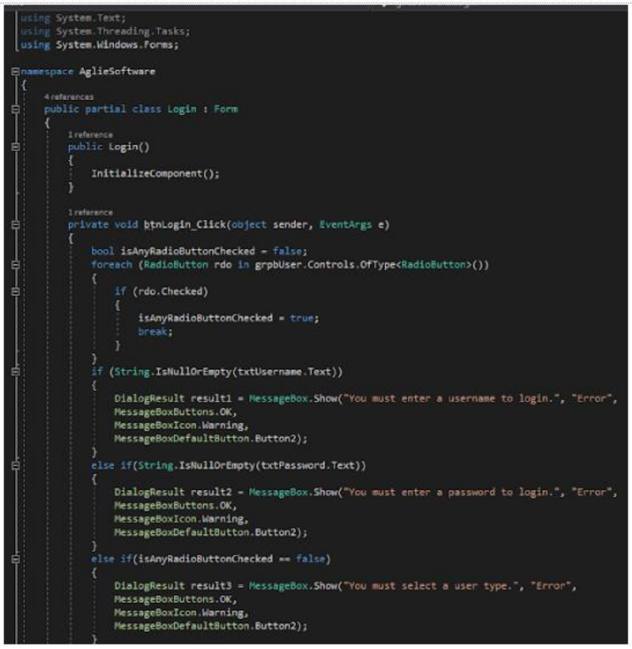
Who are you?

○ Student
○ Module Manager
○ Module Checker

Username:

Password:

Current code

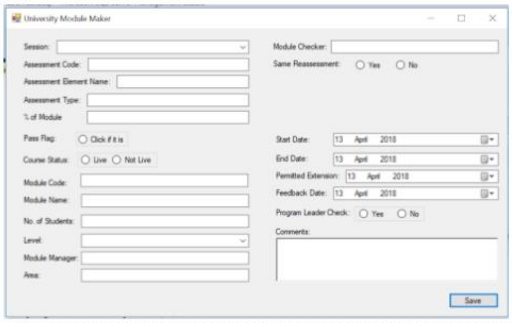


```
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace AglieSoftware
{
    4 references
    public partial class Login : Form
    {
        1 reference
        public Login()
        {
            InitializeComponent();
        }

        1 reference
        private void btnLogin_Click(object sender, EventArgs e)
        {
            bool isAnyRadioButtonChecked = false;
            foreach (RadioButton rdo in grpbUser.Controls.OfType<RadioButton>())
            {
                if (rdo.Checked)
                {
                    isAnyRadioButtonChecked = true;
                    break;
                }
            }
            if (String.IsNullOrEmpty(txtUsername.Text))
            {
                DialogResult result1 = MessageBox.Show("You must enter a username to login.", "Error",
                MessageBoxButtons.OK,
                MessageBoxIcon.Warning,
                MessageBoxDefaultButton.Button2);
            }
            else if(String.IsNullOrEmpty(txtPassword.Text))
            {
                DialogResult result2 = MessageBox.Show("You must enter a password to login.", "Error",
                MessageBoxButtons.OK,
                MessageBoxIcon.Warning,
                MessageBoxDefaultButton.Button2);
            }
            else if(isAnyRadioButtonChecked == false)
            {
                DialogResult result3 = MessageBox.Show("You must select a user type.", "Error",
                MessageBoxButtons.OK,
                MessageBoxIcon.Warning,
                MessageBoxDefaultButton.Button2);
            }
```

Gui


University Module Maker

Session:

Assessment Code:

Assessment Element Name:

Assessment Type:

% of Module

Pass Flag: ○ Click if it is

Course Status: ○ Live ○ Not Live

Module Code:

Module Name:

No. of Students:

Level:

Module Manager:

Area:

Module Checker:

Same Reassessment: ○ Yes ○ No

Start Date: 13 April 2018

End Date: 13 April 2018

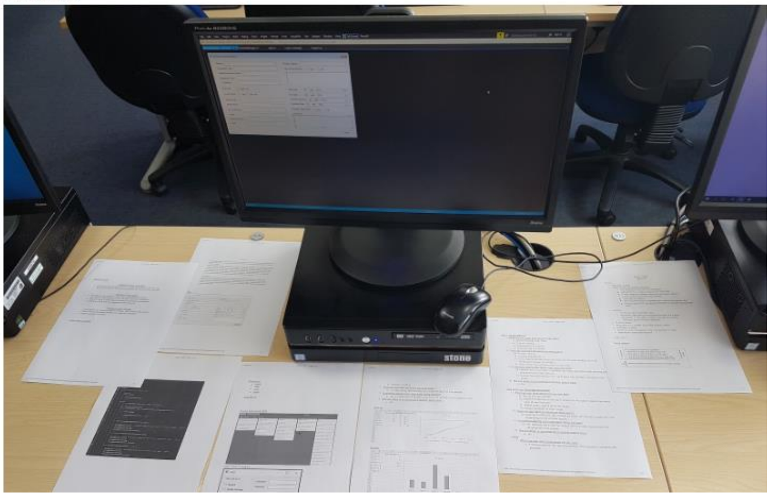
Permitted Extension: 13 April 2018

Feedback Date: 13 April 2018

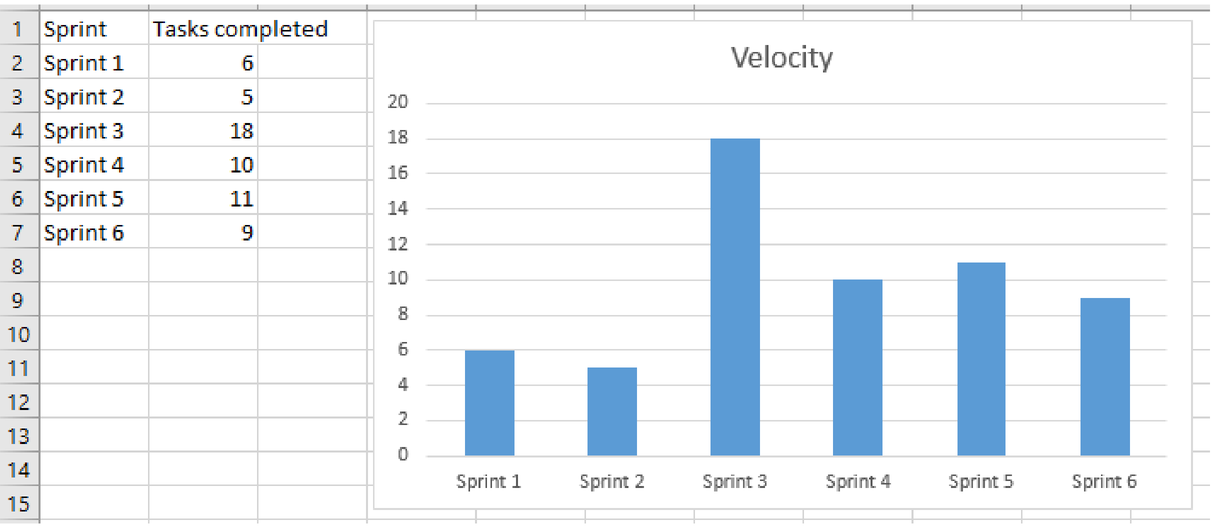
Program Leader Check: ○ Yes ○ No

Comments:

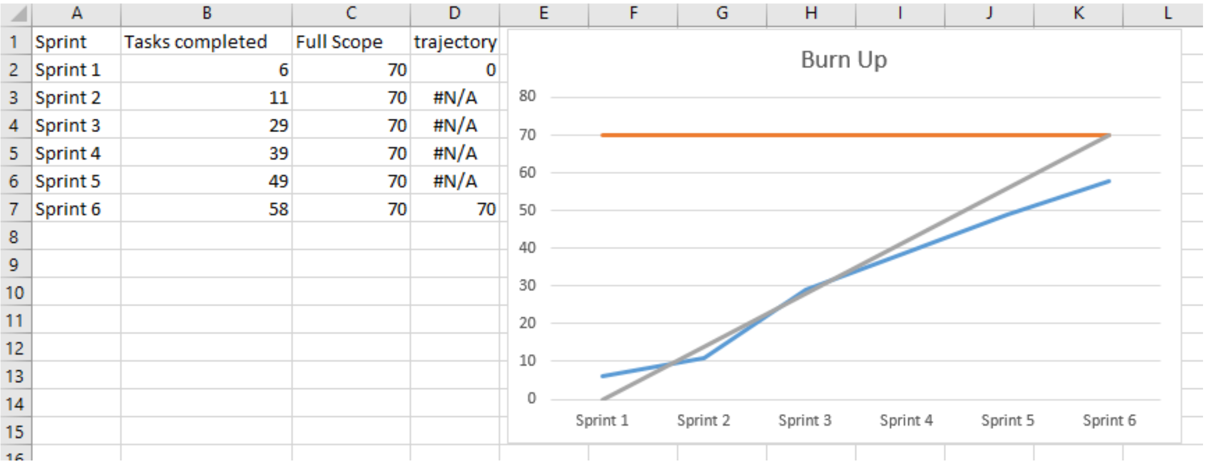
Snapshot 3



Final metrics



	Sprint	Tasks completed
1	Sprint	Tasks completed
2	Sprint 1	6
3	Sprint 2	5
4	Sprint 3	18
5	Sprint 4	10
6	Sprint 5	11
7	Sprint 6	9
8		
9		
10		
11		
12		
13		
14		
15		



Sprint	Tasks completed	Full Scope	trajectory
Sprint 1	6	70	0
Sprint 2	11	70	#N/A
Sprint 3	29	70	#N/A
Sprint 4	39	70	#N/A
Sprint 5	49	70	#N/A
Sprint 6	58	70	70

Scrum 12/04
Sprint 3

Product Backlog

- Functional code
- Create a user interface design for the entire system.
- Portfolio
- Tutorial Video
- Snapshot No 2 (13/04)

Sprint Backlog
Luke, Alin and Neal - Functional coding.
Lewis - Portfolio writing
Ieaun - Time-planning estimation
All - Designs for mock-ups

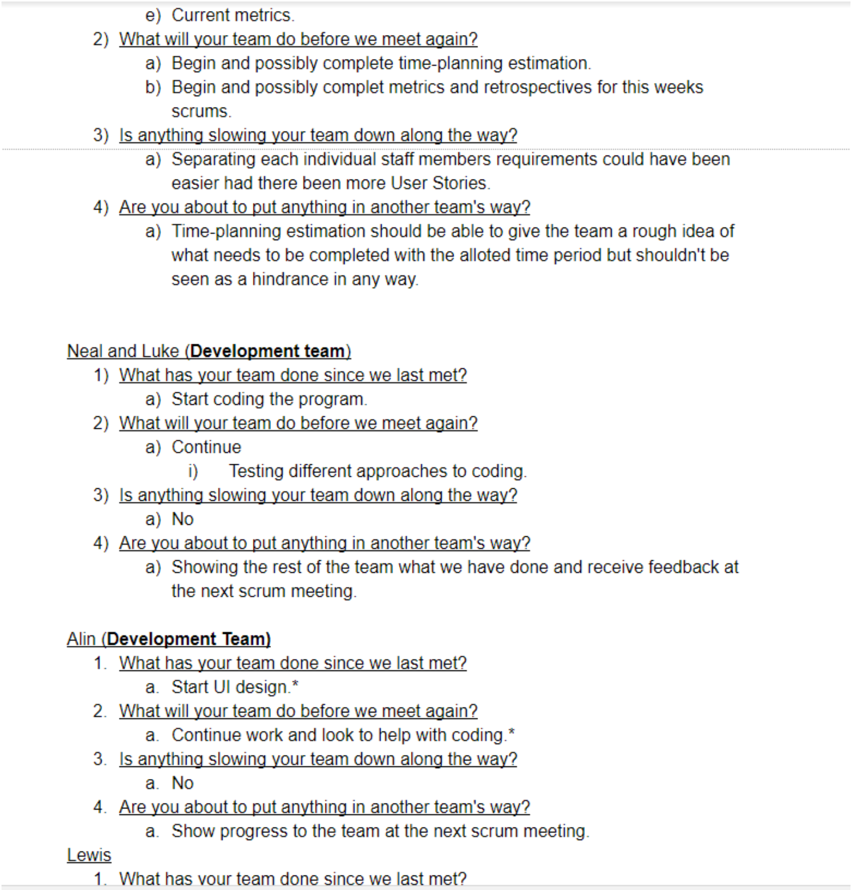
- 1 week sprint

Sprint review

15 minutes	Each participant answers four questions: 1. What has your team done since we last met? 2. What will your team do before we meet again? 3. Is anything slowing your team down on in their way? 4. Are you about to put anything in another team's way?	No personal names
As needed	Resolve problems and discuss issues on the team backlog.	

Ieaun(**Scrum Master**)

1) What has your team done since we last met?
    a) Completed acceptance criteria on all given user stories.
    b) Completed Personas for each User.
    c) Created work backlog file.
    d) Completed Time-Planning estimation.
    e) Current metrics.
2) What will your team do before we meet again?
    a) Begin and possibly complete time-planning estimation.



e) Current metrics.

2) What will your team do before we meet again?
   a) Begin and possibly complete time-planning estimation.
   b) Begin and possibly complet metrics and retrospectives for this weeks scrums.
3) Is anything slowing your team down along the way?
   a) Separating each individual staff members requirements could have been easier had there been more User Stories.
4) Are you about to put anything in another team's way?
   a) Time-planning estimation should be able to give the team a rough idea of what needs to be completed with the alloted time period but shouldn't be seen as a hindrance in any way.

Neal and Luke (**Development team**)

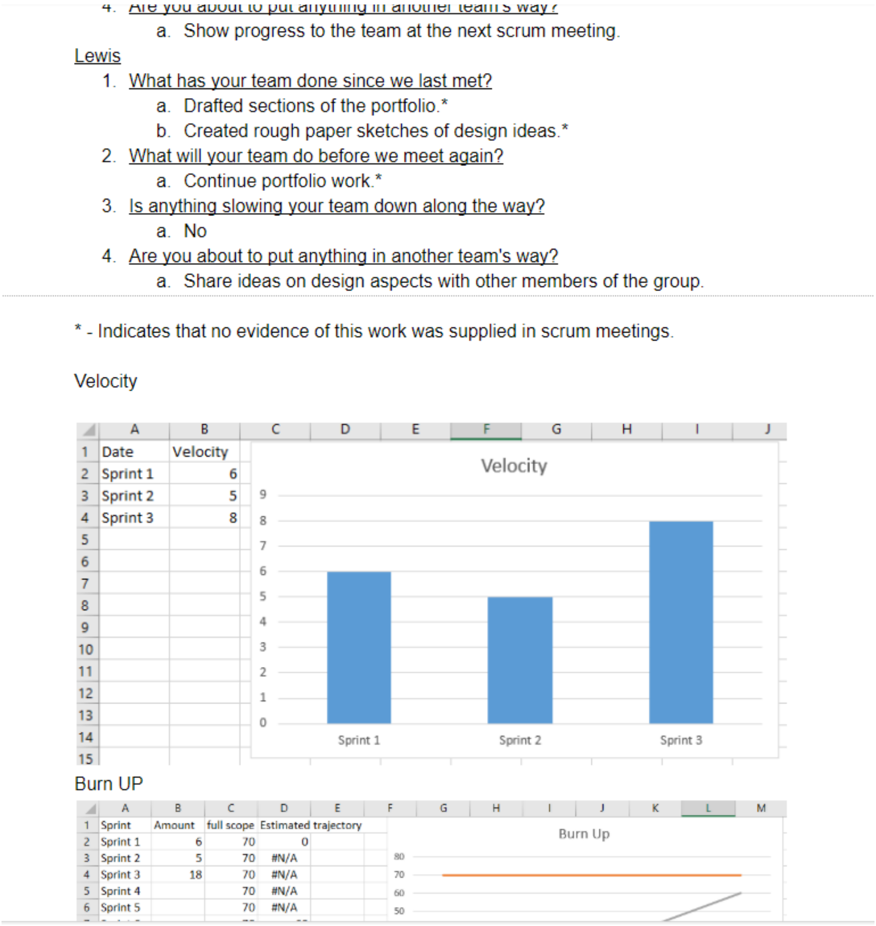
1) What has your team done since we last met?
   a) Start coding the program.
2) What will your team do before we meet again?
   a) Continue
      i) Testing different approaches to coding.
3) Is anything slowing your team down along the way?
   a) No
4) Are you about to put anything in another team's way?
   a) Showing the rest of the team what we have done and receive feedback at the next scrum meeting.

Alin (**Development Team)**

1. What has your team done since we last met?
   a. Start UI design.*
2. What will your team do before we meet again?
   a. Continue work and look to help with coding.*
3. Is anything slowing your team down along the way?
   a. No
4. Are you about to put anything in another team's way?
   a. Show progress to the team at the next scrum meeting.

Lewis

1. What has your team done since we last met?



4. Are you about to put anything in another team's way?
   a. Show progress to the team at the next scrum meeting.

Lewis

1. What has your team done since we last met?
   a. Drafted sections of the portfolio.*
   b. Created rough paper sketches of design ideas.*
2. What will your team do before we meet again?
   a. Continue portfolio work.*
3. Is anything slowing your team down along the way?
   a. No
4. Are you about to put anything in another team's way?
   a. Share ideas on design aspects with other members of the group.

* - Indicates that no evidence of this work was supplied in scrum meetings.

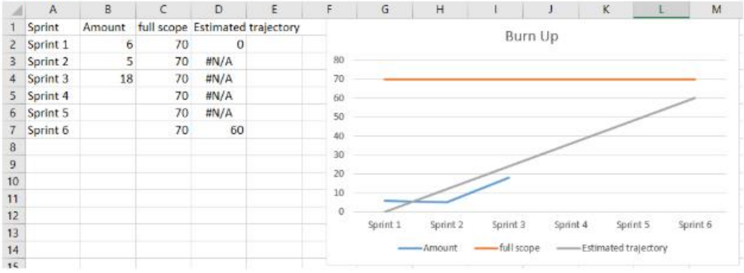
Velocity

Date	Velocity
Sprint 1	6
Sprint 2	5
Sprint 3	8

Burn UP

Sprint	Amount	full scope	Estimated trajectory
Sprint 1	6	70	0
Sprint 2	5	70	#N/A
Sprint 3	18	70	#N/A
Sprint 4		70	#N/A
Sprint 5		70	#N/A

Burn UP



Sprint	Amount	full scope	Estimated trajectory
Sprint 1	6	70	0
Sprint 2	5	70	#N/A
Sprint 3	18	70	#N/A
Sprint 4		70	#N/A
Sprint 5		70	#N/A
Sprint 6		70	60

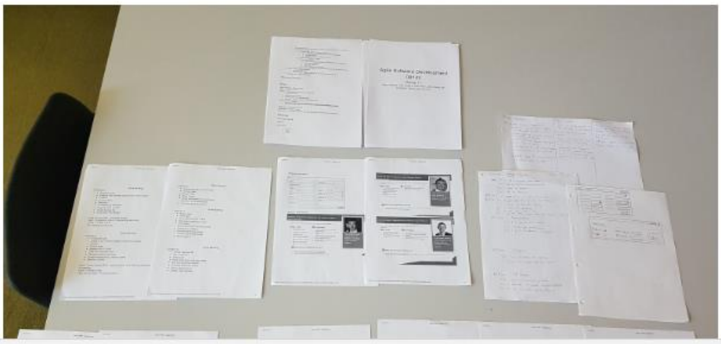
NO of completed tasks: 8

Attendance:

- Ieaun
- Neal
- Luke

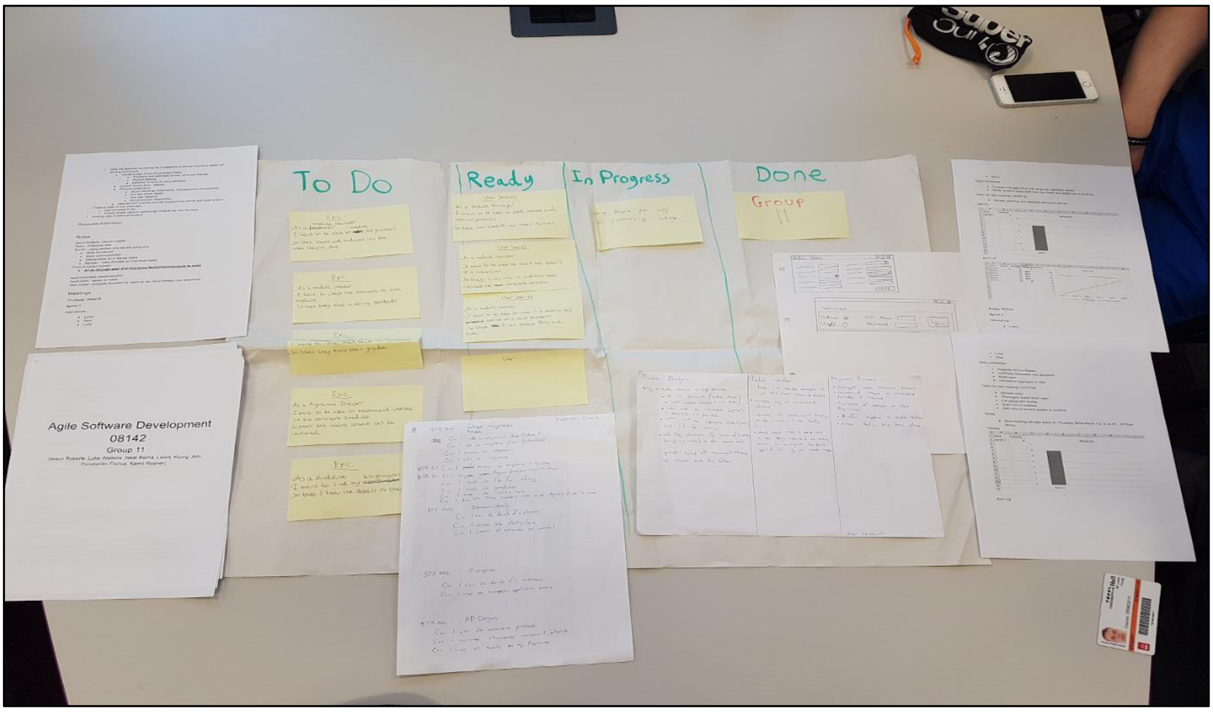
- Ieaun
- Luke
- Neal

Snapshot 2:

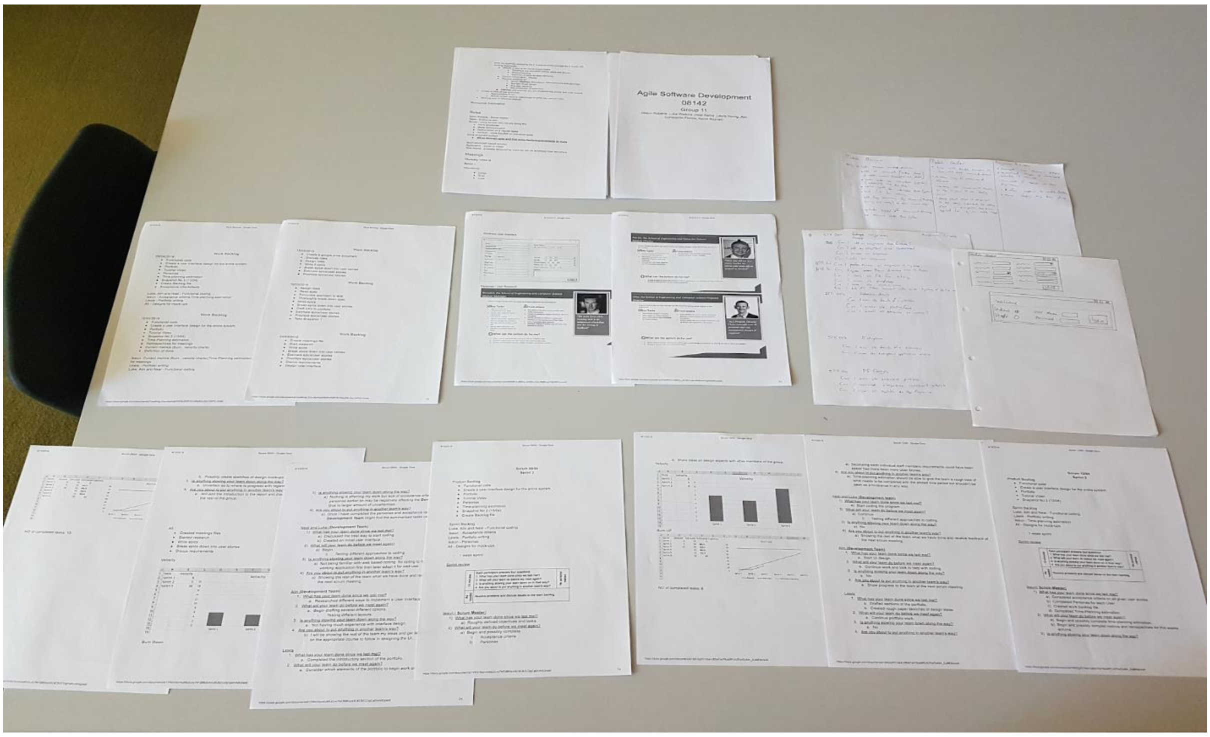


Snapshots

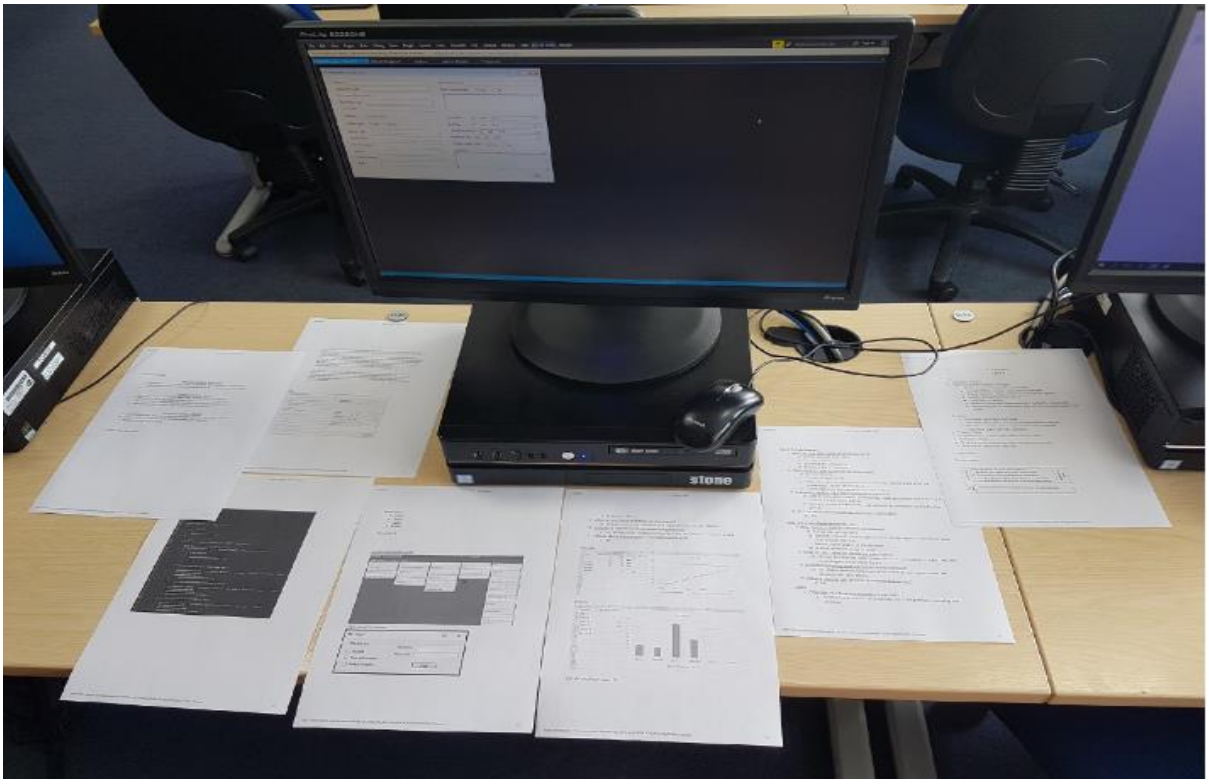
Snapshot(1)



Snapshot(2)



Snapshot(3)



Snapshot(4)