

2013056 Agile assignment 1

by Owen Duncan Skilton

FILE	25302_OWEN_DUNCAN_SKILTON_2013056_AGILE_ASSIGNMENT_1_302864_116187407.DOCX (3.99M)		
TIME SUBMITTED	04-DEC-2020 07:33PM (UTC+0000)	WORD COUNT	3848
SUBMISSION ID	138375800	CHARACTER COUNT	20216

LEVEL 7 ASSESSMENT SPECIFICATION

Student name:	Owen Skilton	Student ID number:	2013056
Programme:	Agile SEP		
Module:	ACCA7001		
Module code:		Contribution to Overall Module Assessment (%):	
Lecturer:	Mike Dacey	Internal Verifier:	
Assignment Title:	Assignment 1	Word count (or equivalent):	3000
Submission deadline:	4/12/20	Return date of provisional marks & written feedback:	
Submission method:	All written assessments, where practical and possible, must be submitted via Turnitin unless otherwise instructed by the Lecturer. (Please DO NOT put this assessment specification into Turnitin or it will match many similarities with other students' submissions.) Alternative submission method (if applicable): Late submission of the assessment will result in a late penalty mark. Penalties for late submission: Up to one week late, maximum mark of 50%. Over one week late, 0%. Only the Extenuating Circumstances Panel may approve a change to submission dates.		
Academic honesty / referencing:	Academic honesty is required. In the main body of your submission you must give credit to authors on whose research and ideas your work is based. Append to your submission a reference list that indicates the books, articles, etc. that you have used, cited or quoted in order to complete this assessment.		

1. Introduction

The DVLA is an organisation within the United Kingdom tasked with creating and managing 49 million driver records [1] which increase every year. With such a large task to undertake and with so many components and departments related to it, it's crucial to be able to effectively manage and handle the tasks required to maintain the integrity of these records. The objective of this report is to evaluate the agile delivery process for the development of software artefacts within the DVLA as well as analysing the tools, methods and techniques utilised by the organisation to effectively and efficiently operate under changing circumstances such as the COVID-19 pandemic as well as onsite collaboration.

1.1 Agile overview & goal

Agile can be summarised as a project delivery methodology whereby several groups of individuals collaborate to design, develop and maintain a product or service whose requirements evolve continuously throughout what is known as the product life cycle. This means the development of such products or services is incremental and iterative, requiring continuous feedback at each step of development from both customer and developers to ensure the efficiency of delivery and flexibility required to be maintained. [2 p.12] Within the agile methodology lie multiple frameworks allowing project managers to select the most efficient and appropriate framework for the product or service being produced. Even though these frameworks differ in their methods and processes they still abide by what is known as the agile manifesto [3 p.11], four key values and 12 core principles that guide the incremental approach to software development.

The goal of the agile methodology is to allow flexibility to both the customer and developers by providing consistent feedback in 2-4 week reviews where the project plan or tasks assigned can be altered depending on what is achieved during this period known as a sprint. If the customer has a change of heart on the direction certain aspects of the project should be taken then they have the option regularly to voice their opinion. This differs from other project management philosophies such as waterfall whose project plan is known to be more regimented as well as the productivity being much lower than agiles, with 13% faster delivery speed and 19% increase in productivity as a team. [4] Flexibility is maintained through the prioritisation of developing working software that can be scaled up in the future throughout the project. This aspect of agile allows for alterations to the project plan or specific task without impacting the future productivity of the project in a serious manner. [5] It also allows clear concise feedback to be provided to the developers to guide the future direction and development of the project.

2. Agile within DVLA

In order to help the DVLA to incorporate this agile methodology they utilise a large amount of different technological software and processes. The software selected is usually specialised in order to support the particular agile framework selected to begin the project, the framework predominantly selected within the information technology service department consists of Scrum however Kanban is utilised on rare occasions depending on the project scope.

2.1 Project preparation

Before a project is able to begin at the DVLA, squads are formed in sizes ranging between 5-8 people. Squads develop different aspects of the project such as the front facing UI of an application or the back-end process of data storage. Squad interaction occurs normally when these processes meet and require input from both ends of the application. Each squad comprises of 3 main roles with support from external roles that are cross team support. Within each squad there is a role known as the product owner. They are tasked with defining the business strategy alongside the customer, as well as creating the product backlog, a detailed list of features and process that the customer wishes to be included. It is crucial that the product owner acts as a medium between the customer and development team when needed. Another crucial role is the Scrum master, a person tasked with ensuring that everyone on the team understands and follows the Scrum practices and framework. They help the team develop the “definition of done”, the work that was completed to create an increment, defined as “a concrete steppingstone toward the product goal”. [6]

The rest of the squad will comprise of the software engineers, the number will depend on the product scope and how much velocity is needed in order to complete the project by the specified deadline. Engineers are tasked with digesting the product backlog into user stories alongside the business analyst to then perform these tasks during the sprints throughout the project.

Alongside the product owner and Scrum master there is a role known as the business analyst. The business analyst is tasked with being the intermediary between the engineers focused on the technical aspects and the business team tasked with the financial aspect. They focus on defining the project requirements from the product owner into tasks that the engineering team can digest and convert into a sprint backlog, an example of which can be viewed in Figure 1. It is crucial for the product owner to understand the customers’ needs and identify the gaps between the customer and development team to facilitate a streamline of communication for both sides. [6]

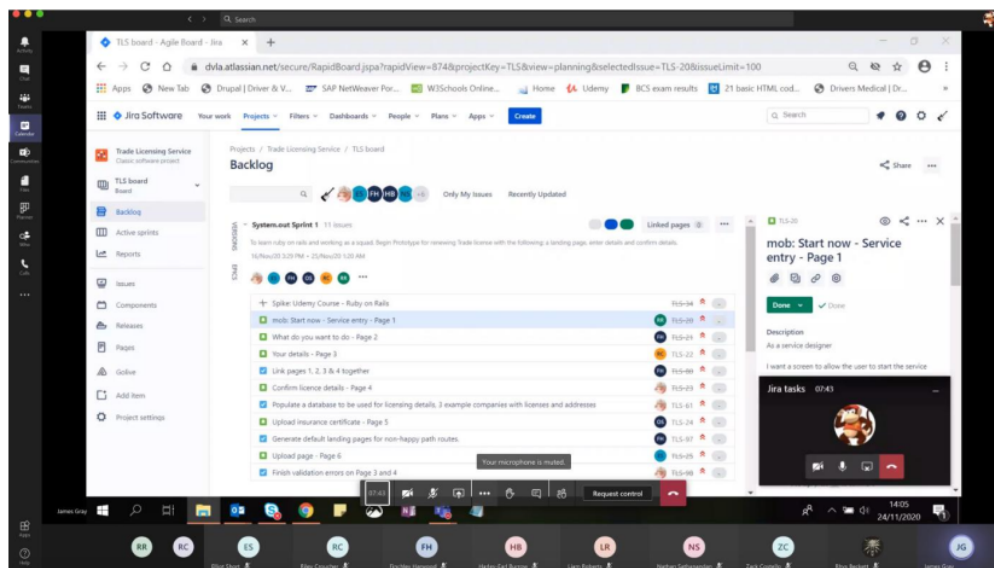


Figure 1, Sprint backlog during the first sprint.

2.2 Sprint start

To begin, the teams attend a sprint planning meeting. This would normally occur in person however at the current time these meetings occur in sprint planning sessions organised on Microsoft teams by the Scrum master. Microsoft teams is an application that allows for virtual meetings to take place allowing users to share their desktop application and messages between each other as if the users were in person, it is a crucial bit of technology while working from home as it allows developers and engineers to perform one of the most crucial part of the agile and Scrum framework which is constant communication, Figure 2 shows this in action. Every team member attends this meeting and the product owner details the scope of the project including the features requested from the customer.

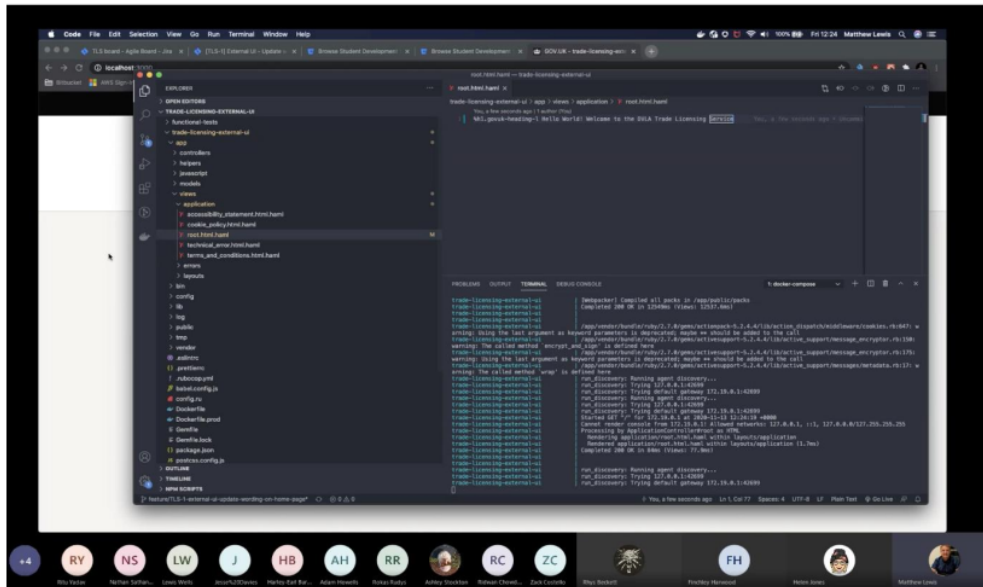


Figure 2, Teams lecture given by a senior developer on Docker.

Within this meeting the team will start digesting the product backlog into user stories that need to be developed in order for the feature to function. A number of techniques can be used to develop user stories including the Moscow technique to evaluate priority as well as forming user stories in the pattern: as a “customer, user” etc, I want... So that... This can be seen in Figure 3. This cataloguing of backlogs is performed on Jira, a piece of software designed for work management and tailored to work under the agile methodology. [7] Jira is used for a multitude of reasons, from creating and storing the product and sprint backlog to interacting with the project’s repository for a more user-friendly experience plus providing detailed feedback reports at the end of sprint to help with the retrospective. In order to know how many user stories to be taken into the sprint backlog, they need to be weighted depending on the amount of time it will take a developer to complete. Jira makes this process very simple as you are able to assign weight to stories and the application automatically calculates the number of days it would take based off of an 8-hour working window.

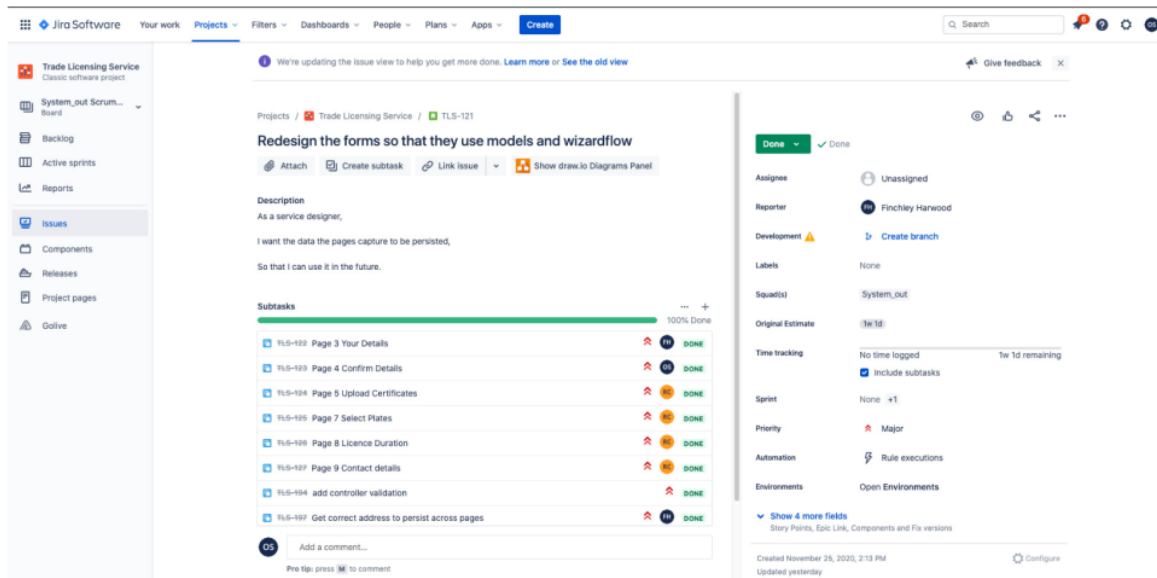


Figure 3, Example of a user story creation.

The team then decides on the number of velocity points to assign to this sprint, usually dictated by the weight of the amount of work completed in the previous sprint, however, if it is a fresh project, the team members usually estimate based off of previous experience. User stories are then taken in totalling the same weight as the team's velocity value. If continuing from a previous sprint, the previous backlog may be taken across and some tasks weight may be adjusted to compensate for the difficulty and amount of work already completed. During my first sprint we elected on a velocity of 200hours for the first sprint with a team of 6 allowing for 6hrs and 40 minutes a day worth of work for each member. If learning is required to perform user stories, a spike in the sprint backlog can be implemented where developers are given a set time to prepare their knowledge for the next sprint. During our first sprint we set aside 2 days for Ruby on rails framework learning. The final task for the teams is to assign a sprint goal, it will normally include at what stage the project will be at as well as mention the changes to the current product, or the new features introduced during that sprint. The project we were assigned to relied heavily on the rails framework so we assigned a sprint goal as seen in Figure 4.

Closed sprint, ended by [Finchley Harwood](#) 16/Nov/20 3:29 PM - 25/Nov/20 2:06 PM [View linked pages](#)

To learn ruby on rails and working as a squad. Begin Prototype for renewing Trade license with the following: a landing page, enter details and confirm details.

Figure 4, sprint goal for the 1st sprint.

2.3 During the sprint

Multiple different technologies are used during the sprint, for software developers, the application they spend most of their time on is a text editor. There are multiple options, some tailored towards certain applications and frameworks and others towards certain languages. During my time at the DVLA the predominant amount of work has been using the ruby on rails framework, which I found Visual studio code was best suited for this as it interacts with Jira and bit bucket very effectively, allowing for easy branch switching and git control within one application. VScode also has access to multiple extensions on the marketplace which can make a developer's life much easier such as autocomplete and colour coordination. Bit bucket is a crucial tool to any web development project, to generalise, bit bucket is an online repository hosting service that allows multiple developers to work on the same project across multiple devices. Bit bucket is incredibly valuable as it has many features including, version control, automated testing interaction and the ability for tracking deployments. Bit bucket also interacts with Jira very efficiently, managing branches throughout the project as well as the commits and merges of individual branches giving an incredibly friendly UI experience to the developer. Figure 5 shows a common ruby on rails bit bucket repository.

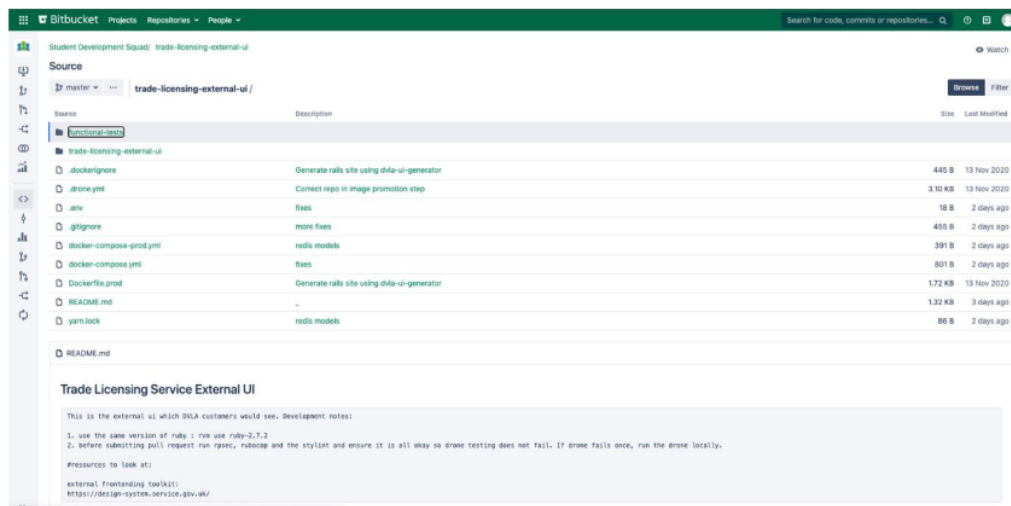


Figure 5, bit bucket repository for the Trade licensing project

A common routine for a developer within a DVLA project is to attend the daily Scrum meeting to discuss yesterday's work and what will be completed that day to add towards the sprint goal. Developers then review the sprint backlogs on Jira and assign themselves work. In some cases, work may be accompanied with a screenshot of the desired aesthetic such as web application or UI. The DVLA utilises a design editor known as sketch to create the models that the developers use as reference. Figure 6 shows a sketch example we received during the TLS project. The developer can utilise the branch creation feature within a Jira user story, this will link this specific user story to the bit bucket repository and ensures any work pushed into it in the future is linked to this branch within the sprint backlog. The developer then needs to ensure that any work they complete takes place within this branch. This can either be done in the command terminal by entering the specific git commands within the local repository for the project, or, if utilising a text editor that allows it, navigate to the Jira and bit bucket extensions and log in with their respected accounts which eliminates the need for a terminal and allows for easier branch switching, this process can be seen in Figure 7. The developer can now complete this user story on their local machine and once complete they have the option to then run predesigned testing tools, these are discussed within the testing section. After testing they can stage and commit these changes before pushing to their branch on the online repository. This local branch then is scrutinised by software known as drone.io, also discussed later, and once passed can then be approved by reviewers to merge into the master branch where other developers then have the option to pull it to their own machine. This process of story and branch completion is the format followed for the remainder of the sprint.

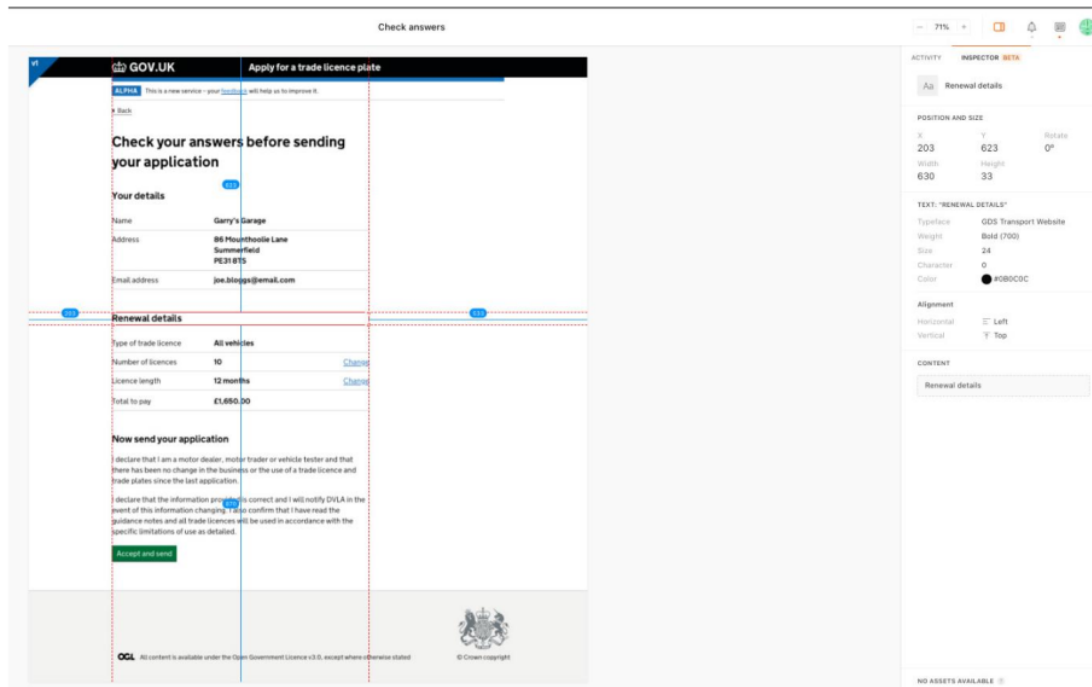


Figure 6, Sketch given from UX design team for a confirmation page.

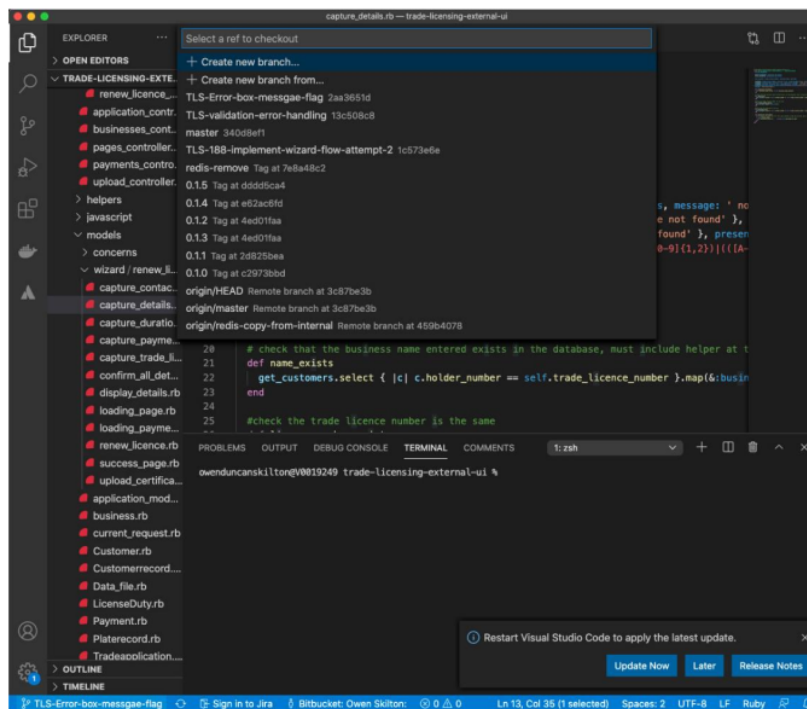
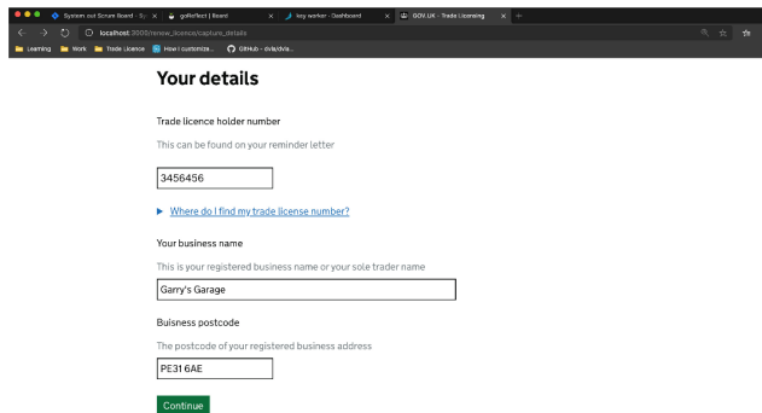


Figure 7, Jira branch switching.

2.4 Sprint review

The sprint review, like the sprint planning, is held once during the sprint at the end with all team members involved where they present the work that was achieved or developed towards the final product, a summation of all the increments, to the customer and stakeholders. The customer / stakeholders will then provide feedback to the work presented and request alterations if needed. The product backlog is then adjusted if necessary and the customer is briefed on the current task backlog. The sprint review is a short meeting usually around a few hours for a 2-week sprint, a common length sprint within the DVLA. The meeting usually takes place in person but is currently taking place online and commonly on teams. A lead developer from the current sprint who is well informed on the work that has been undertaken will present the current status of the Jira board and the work created being hosted on his local machine to the team members and customer and will explain the update product that was created during the sprint time frame, the kind of work presented can be seen in figure 8 and 9.



The screenshot shows a web browser window with the URL `localhost:3000/renew_licence/capture_details`. The page is titled "Your details" and contains several input fields for user information. The fields are: "Trade licence holder number" (with a hint "This can be found on your reminder letter" and a value of "3456456"), "Your business name" (with a hint "This is your registered business name or your sole trader name" and a value of "Garry's Garage"), and "Business postcode" (with a hint "The postcode of your registered business address" and a value of "PE31 6AE"). There is a "Continue" button at the bottom.

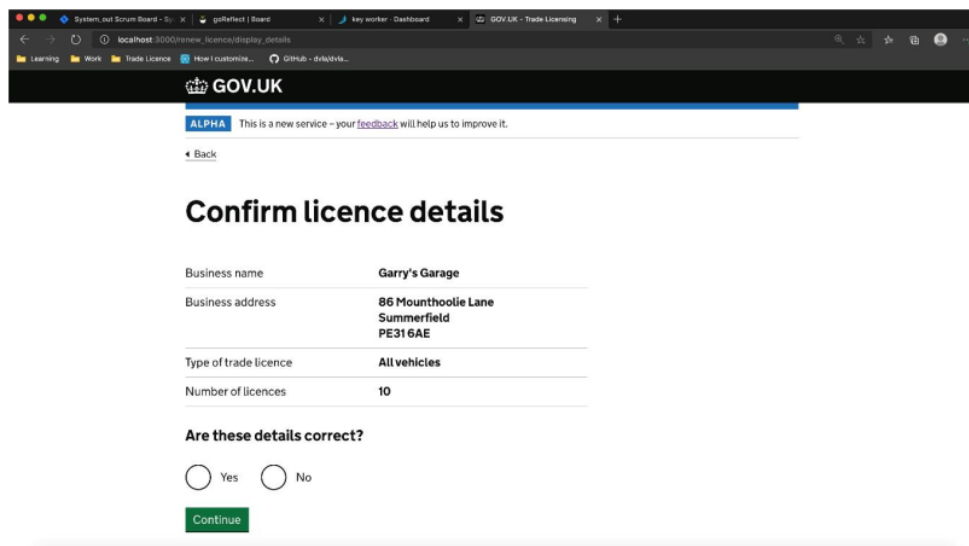
Your details

Trade licence holder number
This can be found on your reminder letter
3456456
[Where do I find my trade licence number?](#)

Your business name
This is your registered business name or your sole trader name
Garry's Garage

Business postcode
The postcode of your registered business address
PE31 6AE
[Continue](#)

Figure 8, Input page from the completed sprint backlog



The screenshot shows a web browser window with the URL `localhost:3000/renew_licence/display_details`. The page is titled "Confirm licence details" and displays the information entered in the previous step. The details are: Business name: "Garry's Garage", Business address: "86 Mounthoolie Lane, Summerfield, PE31 6AE", Type of trade licence: "All vehicles", and Number of licences: "10". There are radio buttons for "Yes" and "No" to confirm the details, and a "Continue" button at the bottom.

GOV.UK

ALPHA This is a new service – your [feedback](#) will help us to improve it.

[Back](#)

Confirm licence details

Business name	Garry's Garage
Business address	86 Mounthoolie Lane Summerfield PE31 6AE
Type of trade licence	All vehicles
Number of licences	10

Are these details correct?

☐ Yes ☐ No

[Continue](#)

Figure 9, display details page for confirmation

2.5 Sprint retrospective

The sprint retrospective differs from the review as its main purpose is for the developers and team leaders to plan on ways to make any future sprints more efficient and effective. This is done by analysing the positives and negatives of the previous sprint just performed and analysing what can be implemented to ensure things do or don't continue in the future. A large focus is oriented towards the reason why problems were not able to be solved, whether it was down to communication lacking or a gap in knowledge, teams will plan for ways to solve this either in future sprint or with the immediate one following. This kind of review is done on a web application known as GoReflect within the DVLA, a site designed to allow users with access to a message board to post anonymously where other users are able to add comments or react to other messages posted, this board can be seen in Figure 10. The message with the most reacts is then discussed more in-depth as a team and made not of for the future sprint. The whole Scrum cycle then repeats after the retrospective which normally takes place on the Friday at the end of the 2-4 week sprint. On the following Monday the sprint planning session usually takes place starting the whole process again until the project is complete.

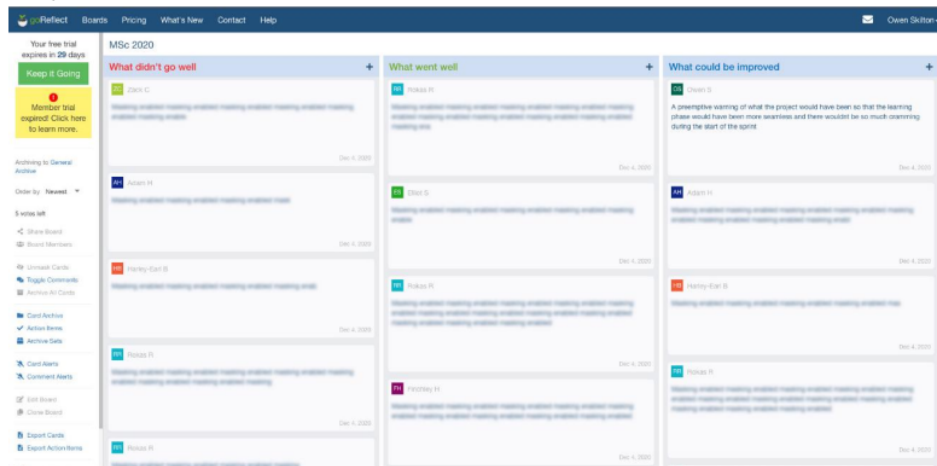


Figure 10, GoReflect board during the final sprint retrospective for the TLS project.

3. Testing (drone.io)

Testing is a vital part of any sort of software development, it ensures the obvious things such as bug fixes and that the code works as intended, but testing can encompass a lot more aspects than just error handling. The DVLA has access to multiple different applications and resources to help increase both the volume and efficiency of testing. A very common form of testing that will occur across languages and applications will be based on the functionality of the code either through compiling or in the case of DVLA web applications, the service will be hosted on the local machine and the developer is able to check multiple things from data entry, validation and navigation. This can be an informal check by simply ensuring data is checked and passed to the correct pages or the navigation links are set up correctly but it's much more difficult if it involves some form of method or function. The DVLA utilises a functional testing tool known as Cucumber, this is a behaviour driven development framework (BDD) usually written in the language Gherkin, a very non-technical, readable language, which is used to write acceptance tests for web applications in ruby on rails [8]. Cucumber is an incredibly efficient way of testing multiple aspects of functions in one script and requires minimal technical knowledge to utilise due to its very readable language Gherkin, during my time in web development Cucumber was utilised to check error message flagging for data entry as well as validation within data fields.

The DVLA enforces a strict coding practice with a large amount of coding standards that can be difficult to remember or spot when a developer is manually reviewing their code, so in order to make the process easier and efficient several tools are utilised. The first tool used is RuboCop, a static code analyser, also known as linter, which essential checks for formatting and grammatical errors in code such as apostrophes instead of speech makes and variable naming. [9] While working with the ruby on rails framework, specific test frameworks are used for testing such as RSpec (Ruby specification). RSpec follows the Given, when, then format for testing the MVC (Model view controller) design pattern. It ensures that data inputted into a form is rejected if it doesn't follow the validation criteria and doesn't save to the model when it fails as well as following the right convention to saving to its named model, interacting with the correct controller and then connects with the correct view when an output is specified. [10]

Once the completed code branch is pushed to the bit bucket repository, a web service known as drone.io utilises Docker containers (packages of code and dependencies that acts like a virtual machine but requires less space [11]) to run prewritten test scripts. The scripts that the DVLA utilise has a reasonably sized focus on linting, similar to the local checks that can be performed with RuboCop. This ensure that the strict coding standards are met for the pushed branches. The rest of the focus for drone is to run a script known as "publish-dry-run". This script performs multiple checks including ensuring that the gems specified for installation exist and the version numbers are correct. The biggest test is to ensure that the ruby web application can successfully exist within a Docker container and be hosted successful in future. This test has a large amount of sub tests to it that the docker application supports that can range from version testing and gem installation all the way to compression and display verification. [12] Every alteration and application that I made during my time with the DVLA has been subject to Drone.io and its predefined script testing. The process usually takes between 7-9 minutes but saves countless hours for the developers across a project's life cycle in comparison. The UI of drone.io can be seen in Figure 11.

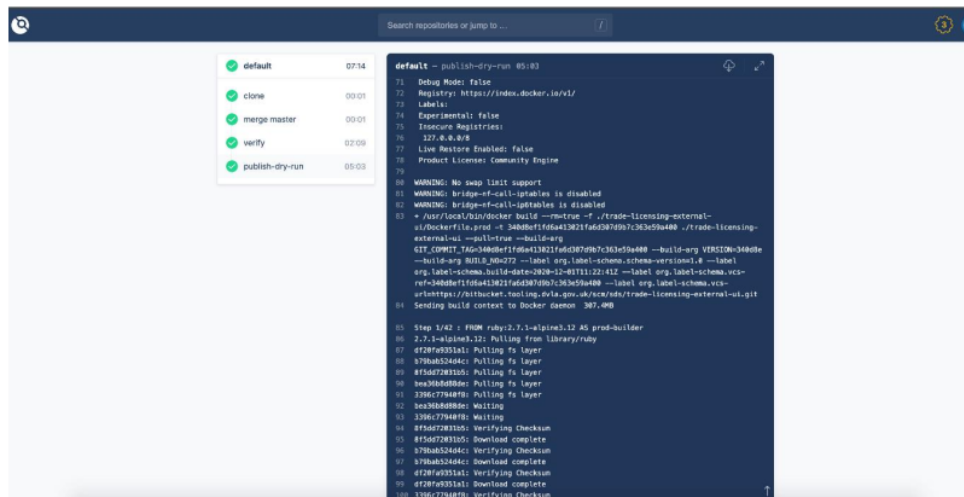


Figure 11, Drone.io testing a branch merge I submitted during the first sprint.

The final stage of a web application is hosting and publishing to the wider world. Though this process is performed by the DVLA cloud team, web developers have the option of using an application known as Spinnaker that is able to facilitate software deployment as well as management. [13] Spinnaker possess its own testing model known as Pipeline. The Pipeline is a sequence of steps that the Spinnaker application performs to ensure a constant and safe way to manage software deployment. These steps are known as stages and can range from infrastructure-based functions or utility scaffolding. Even though our squad never pushed an application through the Spinnaker pipeline it

was useful to be given a lecture to understand the importance of software deployment. Figure 12 shows an example of an application being deployed within Spinnaker.

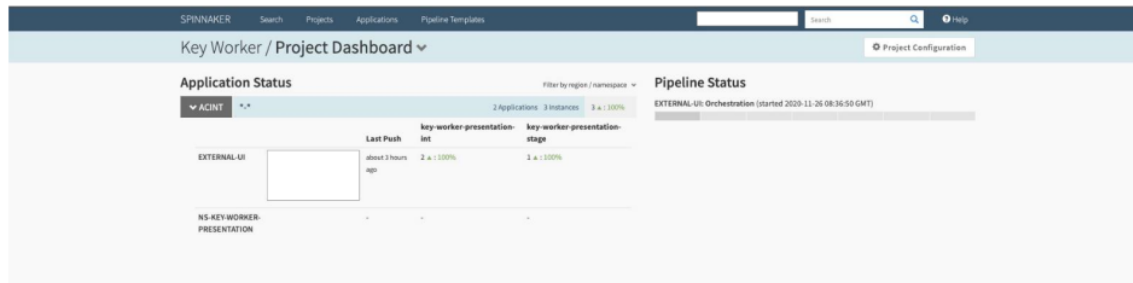


Figure 12, application status within spinnaker.

4. Summary

To summarise the DVLA follows the agile and Scrum framework incredibly closely and only differs in the idea that the customer is the UK citizens using the application and not conventional Stakeholders. A criticism that could be said however would be that the Scrum framework has a large bearing on meetings being attended. This does bare heavily on the amount of time to complete work during the day and is very noticeable in short length sprints where 30% of time is spent discussing work rather than completing user stories.

5. Reference List

- (1) Department of Transport. DVLA Coronavirus (COVID-19) update, 2020. [Online] Available from: <https://www.gov.uk/guidance/dvla-coronavirus-covid-19-update> [Accessed 2 Dec 2020]
- (2) M.A.Badar, A.W.Brown, I.Mistik. Agile software architecture : aligning agile processes and software architectures , 1st edition. Amsterdam: Morgan Kaufmann, 2014, pp.12
- (3) Ibid, pp.11
- (4) B.Davis. Agile practices for waterfall projects : shifting processes for competitive advantage. Plantation, FL: J.Ross Pub, 2013, pp.5
- (5) P.Paquette, M.Frankl. Agile project management for business transformation success, 1st edition. New York: Business Expert Press, 2016, pp.45
- (6) K.Schwaber, J.sutherland. The Scrum Guide. The Definitive Guide to Scrum: The Rules of the Game. 2020 [Online] Available from: <https://www.Scrumguides.org/docs/Scrumguide/v2020/2020-Scrum-Guide-US.pdf#zoom=100> [Accessed 1 Dec 2020]
- (7) Atlassian. What is Jira used for?, 2020. [Online] Available from: <https://www.atlassian.com/software/jira/guides/use-cases/what-is-jira-used-for#jira-for-agile-teams> [Accessed 1 Dec 2020]
- (8) W.Ye. Instant Cucumber BDD how-to a short and quick guide to mastering behavior-driven software development with Cucumber, 1st edition. Birmingham, England: Pack Publishing, 2013, pp.23
- (9) B.Bastov. RuboCop. The Ruby Linter/Formatter that Serves and Protects. [Online] Available from: <https://rubocop.org/#:~:text=RuboCop%20is%20a%20Ruby%20code,community%2Ddriven%20Ruby%20Style%20Gu>

[ide.&text=In%20practice%20RuboCop%20supports%20pretty,of%20the%20problems%20for%20you](#) [Accessed 2 Dec 2020]

(10) The Odin Project. Ruby Programming. Introduction to RSpec. [Online] Available from: <https://www.theodinproject.com/courses/ruby-programming/lessons/introduction-to-rspec> [Accessed 2 Dec 2020]

(11) Docker. What is a Container? A standardized unit of software. [Online] Available from: <https://www.docker.com/resources/what-container> [Accessed 2 Dec 2020]

(12) Drone.io. Overview. [Online] Available from: <https://docs.drone.io/pipeline/overview/> [Accessed 2 Dec 2020]

(13) Spinnaker. Concepts. [Online] Available from: <https://Spinnaker.io/concepts/#application-deployment> [Accessed 2 Dec 2020]

2013056 Agile assignment 1

GRADEMARK REPORT

FINAL GRADE

80/100

GENERAL COMMENTS

Instructor

Agile Process Evaluation (40): 33

Examples (30): 24

Testing (30): 23

Discussion:

Good report with some nice examples provided.

The testing could have added a little more detail on CI but it was still pretty good.

Needed more references distributed throughout the report.

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

