

An Evaluation Of The Agile Development Practices Within The DVLA

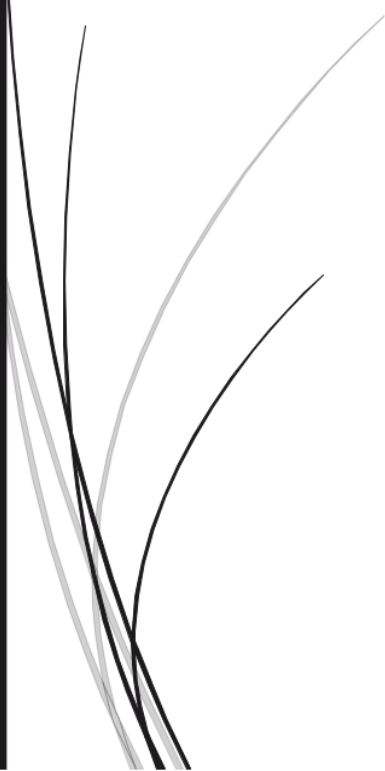
by Nathan Christopher Sathanandan

FILE	25295_NATHAN_CHRISTOPHER_SATHANANDAN_AN_EVALUATION_OF_THE_AGILE_DEVELOPMENT_PRACTICES_WITHIN_THE_DVLA_302864_14417727.PDF (773.58K)		
TIME SUBMITTED	04-DEC-2020 02:17PM (UTC+0000)	WORD COUNT	3901
SUBMISSION ID	138333428	CHARACTER COUNT	21973



12/4/2020

An Evaluation Of The Agile Development Practices Within The DVLA



Nathan Sathanandan
UNIVERSITY OF WALES TRINTY SAINT DAVID

1 Table of Contents

<i>1</i>	<i>Table of Contents</i>	<i>1</i>
<i>2</i>	<i>Table of Figures</i>	<i>2</i>
<i>3</i>	<i>Table of Tables</i>	<i>2</i>
<i>4</i>	<i>Introduction</i>	<i>3</i>
<i>5</i>	<i>Scrum Framework Within the DVLA</i>	<i>4</i>
<i>6</i>	<i>User Stories</i>	<i>6</i>
<i>6.1</i>	<i>What are User Stories?</i>	<i>6</i>
<i>6.2</i>	<i>How User Stories are Estimated</i>	<i>6</i>
<i>6.3</i>	<i>How Jira is Used for User Stories</i>	<i>7</i>
<i>7</i>	<i>Product Backlog</i>	<i>9</i>
<i>7.1</i>	<i>What Is the Product Backlog?</i>	<i>9</i>
<i>7.2</i>	<i>How the Product Backlog is Refined?</i>	<i>10</i>
<i>8</i>	<i>Sprint Backlog</i>	<i>10</i>
<i>8.1</i>	<i>What is the Sprint Backlog?</i>	<i>10</i>
<i>8.2</i>	<i>How Jira is Used for Sprint Backlogs</i>	<i>10</i>
<i>9</i>	<i>Collaboration</i>	<i>12</i>
<i>9.1</i>	<i>Collaboration During COVID-19 Pandemic</i>	<i>12</i>
<i>9.2</i>	<i>The Effectiveness of the Scrum Framework on Collaborations</i>	<i>12</i>
<i>9.3</i>	<i>The DVLA Tools That Help Collaboration</i>	<i>12</i>
<i>10</i>	<i>Continuous Integration (CI) and Testing</i>	<i>13</i>
<i>10.1</i>	<i>Why Testing and CI are Important</i>	<i>13</i>
<i>10.2</i>	<i>Testing Within the DVLA</i>	<i>14</i>
<i>10.3</i>	<i>CI Within the DVLA</i>	<i>14</i>
<i>11</i>	<i>Evaluation</i>	<i>15</i>
<i>12</i>	<i>References</i>	<i>16</i>

2 Table of Figures

Figure 5.1: A diagram that breaks down the process step by step of the scum framework [5].	5
Figure 6.1: An example of a user stories simple sentence structure.	6
Figure 6.2: An example of the DVLA planning poker method within Microsoft teams.	7
Figure 6.3: A screenshot of the Jira interface for Hyperion squad.	8
Figure 6.4: A screenshot of the user story input on Jira.	8
Figure 6.5: A screenshot of the story point feature within Jira.	9
Figure 7.1: A screenshot that gives of the product backlog looks like within Jira.	9
Figure 8.1: A screenshot of a the sprint backlog within Jira.	11
Figure 8.2: A screenshot of the Hyperion squads burn down chart from Jira.	11
Figure 9.1: A screenshot of the differences section within Bitbucket.	13
Figure 9.2: A screenshot of a goReflect board.	13
Figure 10.1: An example of the Gherkin syntax written for a DVLA test.	14
Figure 10.2: A screenshot of drone building and running unit tests.	15

3 Table of Tables

Table 4-1: A table showing what the 4 core values of the agile manifesto.	3
Table 4-2: A table that shows the 12 principles of the agile manifesto.	3

4 Introduction

Over many decades there have been many attempts at creating the perfect project management approach to software engineering. Methodologies such as iterative and incremental development (IID), waterfall, spiral, rapid application development (RAD) and many others have been or are still being used to help software engineers plan, create, test and deploy their products [1]. Then in 2001 a group of 17 people came together to create the agile manifesto, which contained a list of values and principles that supported all the frameworks [1].

Compared to other methodologies the agile approach is more of a mindset or philosophy which is guided by 4 core values and 12 principles [1]. These core values and principles are displayed in table 4.1 and 4.2. below.

Table 4-1: A table showing what the 4 core values of the agile manifesto.

Core value number	Core value description
1 st	Individuals and interactions over processes and tools [2].
2 nd	Working software over comprehensive documentation [2].
3 rd	Customer collaboration over contract negotiation [2].
4 th	Responding to change over following a plan [2].

Table 4-2: A table that shows the 12 principles of the agile manifesto.

Principle number	Principle description
1 st	Our highest priority is to satisfy the customer through early and continuous delivery of valuable software [3].
2 nd	Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage [3].
3 rd	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale [3].
4 th	Business people and developers must work together daily throughout the project[3].
5 th	Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done [3].
6 th	The most efficient and effective method of conveying information to and within a

	development team is face-to-face conversation [3].
7 th	Working software is the primary measure of progress [3].
8 th	Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely [3].
9 th	Continuous attention to technical excellence and good design enhances agility [3].
10 th	Simplicity--the art of maximizing the amount of work not done--is essential [3].
11 th	The best architectures, requirements, and designs emerge from self-organizing teams [3].
12 th	At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly [3].

Currently there are many agile frameworks that exist however, this report will focus on the scrum framework as this is the framework that the DVLA use for their software development projects.

5 Scrum Framework Within the DVLA

The scrum framework is an iterative and incremental which relies on three principles. These three core principles are transparency, inspection and adaptation [4]. There are also 5 main values within the scrum framework, commitment, courage, focus, openness and respect, all of these values help further cement the agile process in the scrum framework [4]. These principles and values help ensure that the agile manifesto is maintained through the process of development.

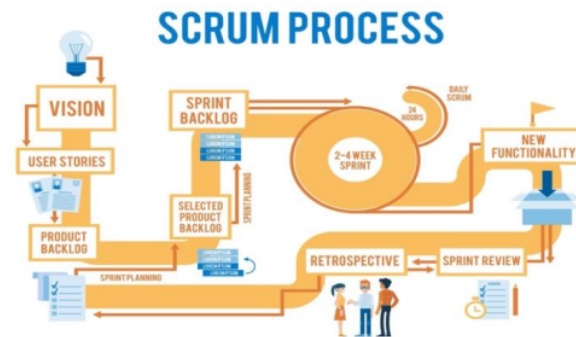


Figure 5.1: A diagram that breaks down the process step by step of the scum framework [5].

Figure 5.1 above gives a good idea of the scrum process. There are three major roles that drive the scrum process, which are the product owner, the scrum master and the development team (better known as a scrum team in the DVLA) [4]. The first step in the scrum process is to create what are called user stories after a project has been approved. Next the user stories are added to the artefact known as the product backlog where all the stories for a project are kept [4]. Once the product backlog is completed stories will be refined and added to the sprint backlog once they have met the definition of ready (DoR) [4]. Once the sprint backlog is complete the scrum team and master hold the first ceremony known as the sprint planning session where stories are split into tasks [4]. Once the planning session is over a sprint will begin. A sprint is a stipulated amount of time (normally 2-4 weeks) to complete the user stories and tasks in the sprint backlog [4]. During the sprint there are daily scrums where the second sprint ceremony takes place known as the daily stand up, this session gives team members a chance to talk to each other every day. When the sprint is complete the third sprint ceremony session is scheduled which is known as the sprint review where the scrum team present their work to show what they have completed. Following this is the 4th sprint ceremony known as the sprint retrospective where the team comes together to discuss what they all think they could have improved on during the sprint. After the sprint and the ceremonies are completed the process is repeated until the project is complete.

6 User Stories

6.1 What are User Stories?

User stories are general explanations of software features and user requirements that are written from the perspective of the end user [6]. The principle of user stories is to communicate how a software feature will provide value to the customer. There're many advantages to using user stories such as helping enable collaboration, stories can lead to creative solutions. They also help to create a steady momentum and the main advantage is they keep the focus on the user [6]. Within the DVLA user stories are structured as simple sentences, for example “As a [user type], I want to [complete some task], so that [a reason]” (an DVLA example is shown in figure 6.1).

Description

As a contact centre clerk

I want to search the the TLS record

So that I can complete the task assigned to me

Figure 6.1: An example of a user stories simple sentence structure.

Within the DVLA user stories have to meet have certain acceptance criteria, which are used to define the boundaries of a user story so that they can be tested once they have been completed. Acceptance criteria are written in Gherkin syntax, which follows “Given <pre-condition>, when <action on\of the system>, then <testable outcome>” [7]. Using this format allows for easy collaboration between the squads, stakeholders, service designers and business analysts because of the descriptive structure [7]. This Gherkin structure also helps make the stories easier to test because Gherkin can be read by Cucumber allowing software test's to be created [7].

6.2 How User Stories are Estimated

Story estimations can be a challenging task as there are a multitude of factors that need to be taken into consideration which can lead to squads having differences in opinions for the size of a story [8]. Estimation of stories helps to clarify how long or difficult a story is to

develop. The user stories are estimated using the Fibonacci sequence to denote the difficulty of a user story, with 1 being the easy and higher numbers being more difficult, within the DVLA. DVLA user stories are estimated within the scrum sprint refinement within a Microsoft teams meeting due to the covid-19 pandemic. Within the sprint refinement session, the user stories are estimated using the planning poker method used by many companies. This process involves the squad looking at a story that needs to be estimated and briefly discuss what the story entails, then each member of the squad will mentally formulate their estimate [8]. Due to the Covid pandemic squads can no longer do this in person, rather now they use the Microsoft teams chat to reveal their number after the scum master has counted down from 3 (figure 6.2 below shows an example of this). If squad members have different numbers, they will explain the reasons for the story point they have allocated and then a mutual story point will be allocated once the team come to an agreement.

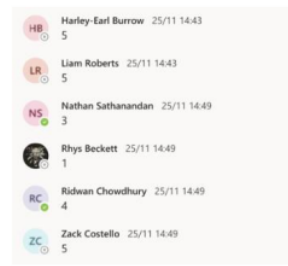


Figure 6.2: An example of the DVLA planning poker method within Microsoft teams.

6.3 How Jira is Used for User Stories

Jira is the main tool in which the DVLA used to create and store user stories for squads, product owners (service designers) and business analysts to see. Figure 6.3 below shows what the Jira platform looks like. Jira is an extremely useful tool as it allows everyone in the DVLA to easily collaborate on stories together.

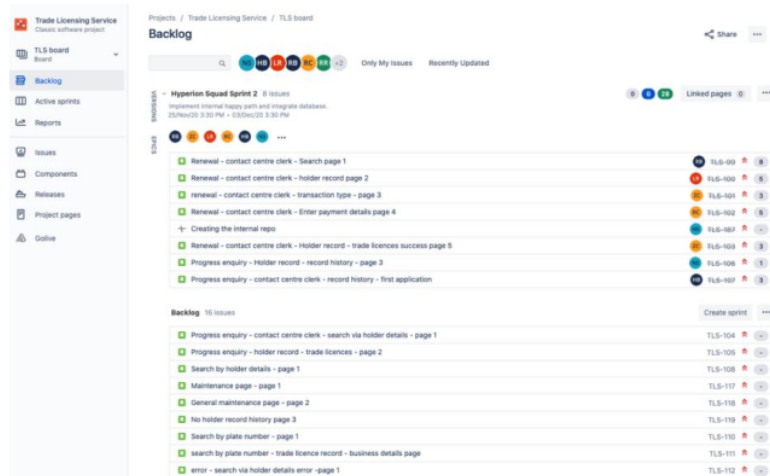


Figure 6.3: A screenshot of the Jira interface for Hyperion squad.

To create a user story in Jira the user has click on the create issue button within Jira which will bring up the screen shown in figure 6.4. This window allows the users to easily add the description of the user story and even add any UX design photos that the service team may have created for the user story.

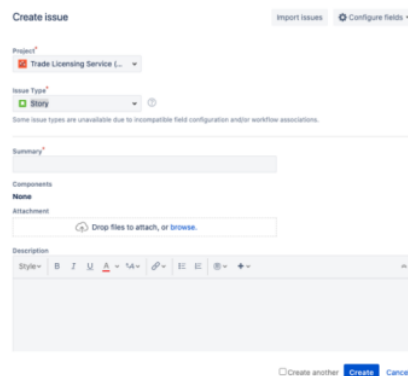


Figure 6.4: A screenshot of the user story input on Jira.

Jira also allows you to enter the story point for a user story during a squad's sprint refinement session. Figure 6.5 below shows where the story point is entered on the Jira interface. The story points are also shown within the sprint backlog that can be seen in figure 6.3 above.

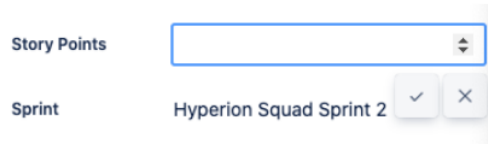


Figure 6.5: A screenshot of the story point feature within Jira.

7 ***Product Backlog***

7.1 ***What Is the Product Backlog?***

The product backlog is where user stories are stored after they have been completed by the product owner [9]. The product owner will also contain epics, which are large requirements that can be broken down into user stories [9]. Product owners have the sole responsibility for breaking down epics into user stories, prioritising user stories and generally managing the product backlog [9]. The product takes the form of a dynamic list within Jira. Figure 7.1 below shows what the list looks like in Jira.

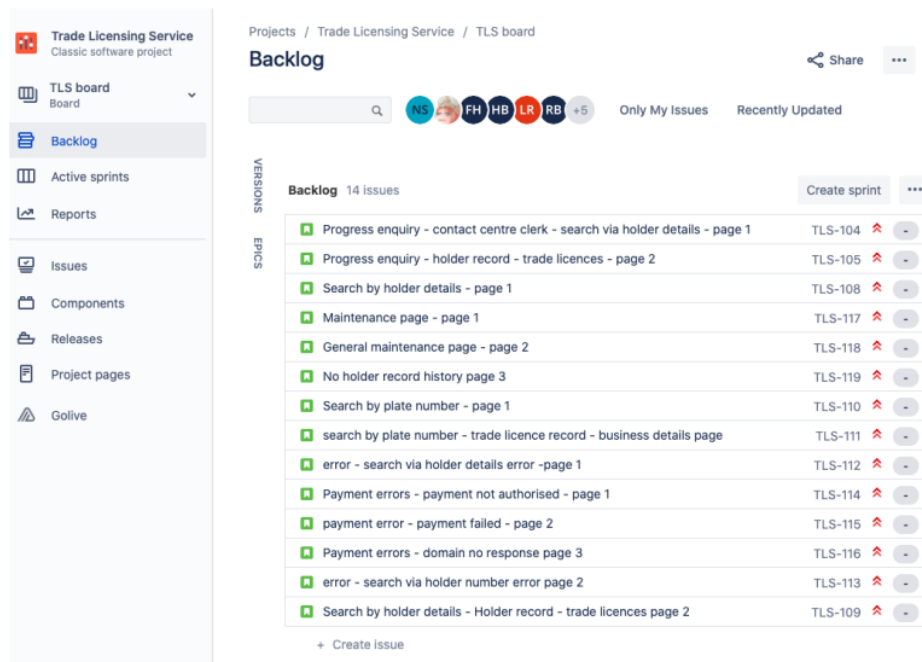


Figure 7.1: A screenshot that gives of the product backlog looks like within Jira.

7.2 How the Product Backlog is Refined?

As mentioned before the DVLA have a scrum refinement session to help estimate user stories, however during this session the squad and product owner will also make sure that the user story meets the definition of ready (DoR) [4]. For a user story in the DVLA to meet the DoR, the team understand enough of the architecture, the story must have been approved by 3 amigos with reference to “3 Amigos Talking Points”; the story must be written as a user story; an acceptance criteria must exist and be understood by team – written in the Gherkin\Cucumber syntax; has no more than 5 acceptance criteria unless agreed as an exception by the team; has been explained by the 3 amigos to the team in refinement, has been estimated by the team, UX sketches exist are created where appropriate for the team and the team understands how to demo the feature. Once a user story has met the DoR it can be considered to be a sprint candidate.

8 Sprint Backlog

8.1 What is the Sprint Backlog?

Sprint backlogs are lists of user stories that a scrum team have agreed to complete during a sprint. In the DVLA the sprint backlog is populated during the sprint planning session which is timeboxed to 2 hours. Squads will plan how they will develop the product increment for the next sprint and then chose the user stories that meet the DoR [4]. Scum teams will also have to choose the highest priority stories available whilst also taking into account the capacity of stories the team can handle. User stories in the sprint backlog will normally also be assigned subtasks, these are technical tasks which make it easier for people to visualise what each user story involves [4].

8.2 How Jira is Used for Sprint Backlogs

Jira is extremely helpful when creating sprint backlogs as users can simply move the user stories from the product backlog straight into the sprint backlog once a sprint has been created (which is shown in figure 8.1).

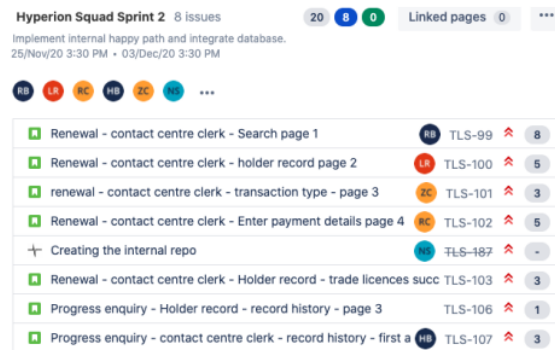


Figure 8.1: A screenshot of a the sprint backlog within Jira.

Subtasks that scum teams agree to add to user stories can also be added on Jira as well. Users can simply click on the user story and click the add sub task button. Users can also assign the subtasks to members in their squad, which means that a sub task doesn't necessarily have to be completed by the person who was assigned to that story. Estimating user stories within Jira also starts to become much easier over time as Jira will produce what's known as a burndown chart during sprints. Burndown charts show the squads how much work is remaining to be completed in the sprint based on how many story points are remaining. These burndown charts become extremely useful as after a squad has been working together for an extended period of time an average burndown velocity (ABV) can be calculated. The ABV allows the squad to see how many stories points they can normally complete during a sprint, and therefore allowing squads to not accept more user stories than they can handle into the sprint backlog. This feature in Jira therefore allows squads to also estimate stories much more accurately as well because members become more familiar with what everyone can handle. Figure 8.2 below shows an example of a burndown chart created during a sprint for Hyperion squad during their sprint.

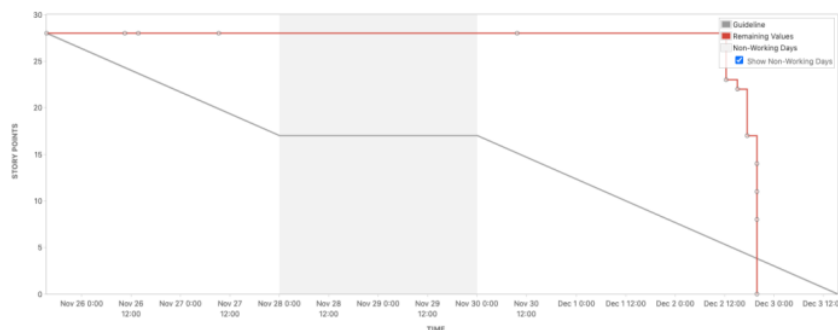


Figure 8.2: A screenshot of the Hyperion squads burn down chart from Jira.

9 Collaboration

9.1 Collaboration During COVID-19 Pandemic

Within the DVLA's process of developing and producing software there is constant collaboration through the use of the scrum framework ceremonies, bitbucket, Jira and Microsoft teams. Microsoft teams has become the pivotal piece of software to allow DVLA staff to collaborate due to the current COVID-19 pandemic which has meant no staff have been allowed to use the DVLA offices. However, despite this the DVLA seem to be working as effectively as ever, it could even be said that the situation has improved the collaboration. This is due to meetings no longer being limited to room sizes, therefore allowing more people who may have useful input to join team meetings.

9.2 The Effectiveness of the Scrum Framework on Collaborations

Processes within the scrum framework such as the sprint refinement, sprint planning, daily stand ups, sprint review and the sprint retrospective also strongly help the DVLA collaborate together seamlessly. All of these processes involve either the scrum team collaborating with each to help complete their increments; the 3 amigos (product owner, dev and tester) working with scrum teams to refine stories; stakeholders and product owners giving feedback from sprint reviews on a new increment the scrum team has produced; the scrum master assisting a member of the scrum team when they are facing a blocker or even the scrum team members giving feedback on what the squad could improve next sprint. The scrum framework is truly a great assist in terms of helping the DVLA collaborate together.

9.3 The DVLA Tools That Help Collaboration

There are also many tools that the DVLA utilise that help ensure that people in the DVLA are collaborating well with one another. The first software has been already been demonstrated at how effective it is at helping with collaboration, the software is called Jira. The second tool is called bitbucket which is an online repository, bitbucket requires at least two members from your scrum team to approve merge requests before changes are implemented into the masters. The people that approve are required to check the changes you've made to the code using the differences section in bitbucket, this process allows users to

make sure each other's code is too the highest standards and also avoid merge conflicts. Figure 9.1 below shows gives an example of what the difference section looks like in bitbucket.

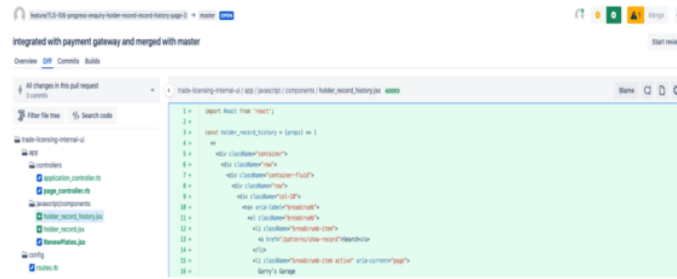


Figure 9.1: A screenshot of the differences section within Bitbucket.

The last main software that helps with collaboration is called goReflect. This software helps squads vote on what they think can be improved on their next sprints and what action items they should focus on in the next sprint to ensure they always improve. Figure 9.2 below gives an example of what the goReflect board looks like.

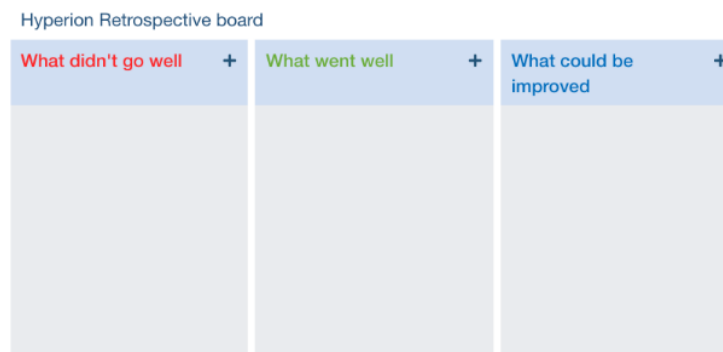


Figure 9.2: A screenshot of a goReflect board.

10 Continuous Integration (CI) and Testing

10.1 Why Testing and CI are Important

The DVLA takes CI and testing very seriously. They take CI seriously as by applying CI tools and practices to your process helps avoid huge conflicts for software engineers which can cause huge problems when deploying new products [10]. Testing is taken seriously because

without, it becomes easy to miss mistakes in our code which could lead to the DVLA having to spend a lot of money to repair those mistakes.

10.2 Testing Within the DVLA

When writing tests, the DVLA take a behaviour driven development (BDD) approach because it helps close the gap between businesspeople and technical people as it encourages collaboration across roles to develop shared understanding of the problem that needs to be solved [11]. BDD also works well for rapid small iterations allowing the process to work well with user stories and gain quick feedback to allow for effective workflow [11]. Using the BDD approach testers use cucumber as it can read the specification which are written in BDD style [7]. Cucumber will read executable specifications which are written in plain text and validates that the software does what the specifications say (Figure 10.1 below is an example of the structure).

```
Scenario: AC-0 return a list of all mt todos  
  
  Given the user has created todos  
  When the uses requests a list of todo items  
  Then the API will return an "OK" response  
  And the user will recieve a list of items
```

Figure 10.1: An example of the Gherkin syntax written for a DVLA test.

For cucumber to understand scenarios they have to follow the basic gherkin syntax, like the example above. Gherkin has grammar rules that make plain text structure enough for cucumber to understand [7]. Gherkin is great as its non-technical, it's easy for anyone to read and can help enforce firm, unambiguous requirements. This method of testing is extremely agile friendly as it allows close collaboration with business owners and developers, the process also helps deliver working software frequently and it enhances the technical excellence within the process [7].

10.3 CI Within the DVLA

Within in bitbucket there is a tool that is linked known as drone which greatly assists with continuously building, testing and deploying code. Drone operates within a docker container and executes stage builds within their own container at each stage of the build and once the build is completed the containers shutdown. The huge advantage to using docker is

that it allows users to quickly make changes to their software and quickly integrate those changes whilst helping avoid major merge conflicts. Figure 10.2 below shows a great example of what a drone build process looks like.

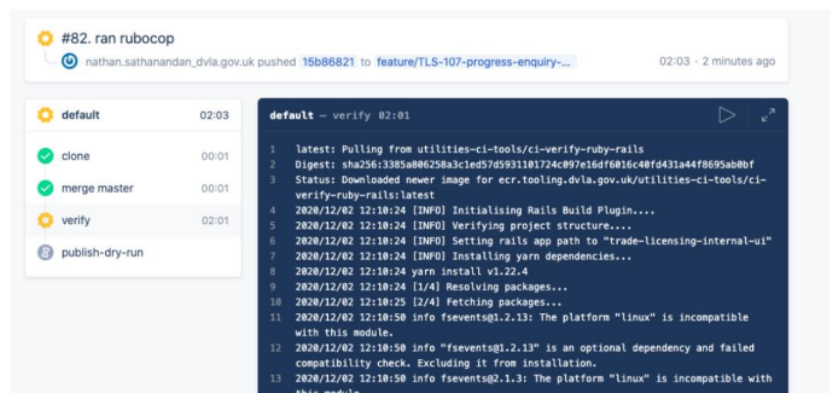


Figure 10.2: A screenshot of drone building and running unit tests.

When drone fails to complete a build it's actually very easy to decipher where the build went wrong and correct it, as you can simply look for the error message within in drone and then correct the mistake within your code.

11 Evaluation

Overall, the DVLA have clearly invested a lot of time, energy and money to ensure that they uphold their agile development practices. They have used a multitude of tools to achieve this (such as Bitbucket, Jira, Microsoft teams and goReflect), however the use of the agile framework known as scrum has definitely helped them a lot at maintain agile practices. For example, using the user stories has helped them maintain the 1st, 2nd, 6th, 9th and 10th principles and the 1st and 2nd core values within the agile manifesto; The estimation processes have ensured that the 3rd, 5th, 6th, 7th and 10th and the 1st core value in the agile manifesto is followed and the scrum ceremonies and artefacts have ensured that every single principle and core value within the agile manifesto is followed religiously within DVLA squads.

12 References

- [1] - Hohl P, Klünder J, van Bennekum A, Lockard R, Gifford J, Münch J, Stupperich M and Schneider K 2018 Back to the future: origins and directions of the "Agile Manifesto" – views of the originators *Journal of Software Engineering Research and Development* 6
- [2] - Beck K, Beedle M, Bennekum A, Cockburn A, Cunningham W, Fowler M, Grenning J, Smith J, Hunt A, Jefferies R, Karen J, Marick B, Martin R, Mellor S, Schwaber K, Sutherland J and Thomas D 2001 Manifesto for Agile Software Development *agile manifesto*
- [3] - Beck K, Beedle M, Bennekum A, Cockburn A, Cunningham W, Fowler M, Grenning J, Highsmith J, Hunt A, Jeffries R, Kern J, Marick B, Martin R, Mellor S, Schwaber K, Sutherland J and Thomas D 2001 Principles behind the Agile Manifesto *Agilemanifesto.org*
- [4] - Measey P 2015 *Agile foundations* (Swindon, UK: BCS Learning & Development)
- [5] - Task 2019 *Scrum Process*
- [6] - Rehjopf M 2020 User Stories | Examples and Template | Atlassian *Atlassian*
- [7] - Anon 2019 Introduction - Cucumber Documentation *Cucumber.io*
- [8] - Radigan D 2020 What are story points and how do you estimate them? *Atlassian*
- [9] - Anon 2020 What is a Product Backlog? *Scrum.org*
- [10] - Boloz D 2020 *IMPROVING USE OF BEHAVIOURDRIVEN DEVELOPMENT IN WEBSPPHERE COMMERCE PROJECTS* Bachelors (Jamk university of applied sciences)
- [11] - Anon 2020 Behaviour-Driven Development - Cucumber Documentation *Cucumber.io*

An Evaluation Of The Agile Development Practices Within The DVLA

GRADEMARK REPORT

FINAL GRADE

64/100

GENERAL COMMENTS

Instructor

Agile Process Evaluation (40): 26

Examples (30): 19

Testing (30): 19

Discussion:

Generally the Agile discussion is quite good but a little more discussion of the DVLA process could have been provided

Testing could have provided more background research of CI testing.

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17
