|  |
| --- |
| University of Dundee |
| AC31005 – Agile Programming |
| Shares project |
|  |
| **Team Jazz** |
| **11/2011** |

|  |
| --- |
|  |

**AC31005 – Agile Programming Assignment**

Jass Shares

## Introduction

This documentation will discuss in detail all of elements that the team has covered during this sprint. It will cover all the planning materials used; the quality assurance steps taken; and our use of source control.

***Please note: All materials referenced in this document can be obtained from*** [***https://github.com/ndzbarr/Jazz/***](https://github.com/ndzbarr/Jazz/) ***unless stated otherwise.***

## Planning Materials

Project Board

Our team’s project board was used as a hub for all the tasks required during this sprint. Each team member was responsible for updating their area of responsibility, providing a snap shot to the team on which tasks remained outstanding, and the tasks that that required starting. This project board was also used as a focal point for our daily scrums. A photograph of the project board is available on git hub.

Sprint Backlog

The teams sprint backlog detailed all the tasks that were required on the project, the amount of effort required, and the team member responsible for each task. Each team member had the responsibility to update the backlog at the beginning of each day, detailing how many hours of effort they believed would be required until the task was considered complete. This allowed the scrum master to see at an instant if/when the project will be completed, and would allow the team to invoke contingency plans if necessary. The backlog is held in excel format, and is available on gitHub

Sprint Burndown Chart

Using the sprint backlog, the scrum master was able to create a burndown chart of the data. This was a graphical representation of the sprint backlog, and allows the team to see the projects expected completion date. The image for the burndown chart is available on gitHub

#### Design of Interface

A simple design sketch for the interface of the project has been included and is available on gitHub. These sketches were used in the final layout of the application.

#### Record of Velocity

The team’s velocity from the previous sprint was taking into consideration when deciding with the customer which user stories could be completed in this sprint. The team’s last sprint velocity was recorded at 12 user story points (USP). However during our second sprint the velocity was only recorded at 10USP. The reason for the lower figure, is this was the maximum USP the team could use on the user stories without going over our previous recorded velocity of 12USP.

#### Sprint Review

From the first sprint review, the team took on board the feedback from the customer to make minor improvements to the layout of the application, specifically, how monitory values were displayed to the customer. Since the changes were minor, and did not require any recordable effort, these were promptly completed and demonstrated to the customer on the second sprint review. Planning the second sprint review was easier than planning the first, since it was possible to draw upon experiences and feedback from the first sprint review.

For the teams second sprint review, demonstrating all the boundary values specified by the user stories was one of the team’s priorities. The team believed that being able to demonstrate that the software would react in the way the customer expected, would provide the application with credibility, whilst allowing the customer to have confidence in the product.

#### Sprint Retrospective

From the first retrospective, the team attempted to incorporate all the improvements that were discussed. It was believed that more interaction with the customer was required, to ensure that the final product would deliver to expectations. The customer was consulted with interface design sketches, acceptance tests, and to seek clarity on specific aspects of the user stories. The team also attempted to seek a new working environment, since during the previous sprint the team considered the environment disruptive. However after consultation, it was agreed that the team would remain in the current environment, since it had all the facilities the team required.

## Quality Assurance

Pair Programing

During the second sprint, the team adopted a paired programming approach to the sprint. Initially, the team struggled to adjust to the new conditions, since it was deemed “disconcerting” to begin with. However with perseverance, it became quickly apparent that it was of benefit to accept and adopt this approach. The team found that errors were less frequent, and code quality improved as a result. However, development time did not suffer. Videos of the team’s pair programming will be available on CD format.

Unit Testing

Unit Tests for this sprint were originally written in Junit4. However after writing the tests it was discovered that Junit4 was difficult to integrate with android projects in the eclipse. All the tests were subsequently rewritten in Junit3 syntax. This allowed the unit tests to integrate seamlessly, and permitted automation. The unit tests were invoked during each build, and allowed a testing history to be established. All the unit tests are available on gitHub under SharesRunTest

#### Metrics

The team used DJUnit in the project to view the metrics of the project. Only one metric was focused on throughout the sprint, the McCabe Cyclomatic complexity. This allowed the team to monitor the available control paths the application could take, and was used to aid in refactoring the program. Although the final product caused the metric to be high, the team deemed the value to be acceptable, in this instance.

Blackbox Testing

Blackbox testing was used by the team to define the testable limits of the application during this sprint. These tests were frequently changed, as the team gained more knowledge on the behaviour of stocks and share pricing. The documentation of this testing is available on gitHub.

Acceptance Tests

Acceptance tests were carried out by the team at the very beginning of the sprint. These were used to form an agreement with the customer on what was deemed acceptable. The acceptance tests were used as a reference point the team would frequently revisit when developing the application, to ensure the final delivered product would be accepted. During initial meetings with the customer, changes were made to the acceptance tests, which were subsequently incorporated. The acceptance tests for this sprint are available on gitHub in the [Project Files Sprint Backlog and Tests/](https://github.com/ndzbarr/Jazz/tree/master/.Project%20Files%20Sprint%20Backlog%20and%20Tests) folder.

Refactoring

Refactoring was carried out by the team periodically. It was agreed that refactoring of the project would be performed after each successful unit test, when changes were conducted, or when any new functionality was added. Samples of the team’s refactoring are available on gitHub in the [Project Files Sprint Backlog and Tests/](https://github.com/ndzbarr/Jazz/tree/master/.Project%20Files%20Sprint%20Backlog%20and%20Tests) folder.

## Source Control

#### DropBox

During the first sprint the team used DropBox as the method of source control. This contained all the project builds as well as all the graphics and documentation the team created. This system was used since some team members could not get github to successfully integrate with the project. This system is no longer used by the team as its source control; however access to this can be given upon request.

#### GitHub

GitHub was adopted by the team during the second sprint to make use of its version control. This allowed team members to work in unison, and aided the merging of the various versions from each team member. This improved development time, and allowed the team to recover previous versions when builds were unsuccessful. The entire project is available from [***https://github.com/ndzbarr/Jazz/***](https://github.com/ndzbarr/Jazz/)***.***