**The Green Git Inspector**

**Project Plan**

**FIT2100 Semester 2, 2019**

**Min Hee Cho, Jack Thomas Whelan, Jun Yoong Ooi, Abhijeet Mondal**

**TEAM green**

**This document contains the team policies that will be followed throughout the duration of the project.**

**Contents**

[**Vision Statement**](#_qu7ndi7nl6ps) **2**

[**Our Team**](#_9hwb5i7tzj92) **2**

[**Process Model: Scrum**](#_lwkj0l4iezai) **3**

[Sprint planning](#_68v86kn86ud0) 3

[The Sprint](#_908gsixcbsm7) 3

[The Daily Scrum](#_sbhjk3srqa50) 4

[Product reviews](#_ki6yr9ktffft) 4

[Retrospectives](#_qhdcr7s4qmoa) 4

[Backlog refinement](#_vfrudfsacmf8) 4

[**Organisation**](#_vfg2jrsr1afo) **5**

[Task allocation](#_xj6138fi9r6e) 5

[Time tracking](#_iegua18mr3vw) 5

[**Definition of Done**](#_dtvomwa2oyvk) **6**

[Checklist for implementing a feature](#_4x6gyuf0b8q9) 6

# 

# Vision Statement

*“For teachers and assignment markers, who must fairly mark students based on their project contributions, The Green Git Inspector Is a GitHub repository inspector, that provides easily accessible contribution statistics. Unlike the competitors, our product is user friendly and calculates all necessary information automatically.”*

# Our Team

|  |  |  |
| --- | --- | --- |
| **Name** | **Contact Details** | **Roles and Responsibilities** |
| Jack Thomas Whelan | Student ID: 29681758  Email: [jwhe0002@student.monash.edu](mailto:jwhe0002@student.monash.edu) | Development Team - takes on and determines how to deliver chunks of work in frequent increments |
| Min Hee Cho | Student ID: 28658736  Email: [mcho0025@student.monash.edu.au](mailto:mcho0025@student.monash.edu.au) | Product Owner - determines what needs to be done and sets the priorities to deliver the highest value |
| Jun Yoong Ooi | Student ID: 29685397  Email: [jooi0012@student.monash.edu](mailto:jooi0012@student.monash.edu) | Scrum Master - protecting the Scrum process and preventing distractions |
| Abhijeet Mondal | Student ID: 29381495  Email: [amon0005@student.monash.edu](mailto:amon0005@student.monash.edu) | Development Team - takes on and determines how to deliver chunks of work in frequent increments |

# Process Model: Scrum

For this project, the Scrum process model will be employed. Scrum is an agile management methodology which aims to produce new capabilities every week (depending on the length of the sprint). Scrum consists of multiple phases within the span of project development.

## Sprint planning

Before each sprint, our team will hold a meeting in which the Scrum Master will facilitate. The Product Owner will clarify the requirements set by the client and prioritise what needs to be done. In this meeting, our team will decide which set of tasks to complete during the sprint and allocate tasks accordingly.

To guarantee the most efficient and productive sprint effectively, the Scrum Master will ensure the level of work to be done is sustainable by the whole team and if necessary, tactfully discuss with the Product Owner to agree to a more feasible outcome.

## The Sprint

Conventional Scrum Sprints are usually a time-box of one month, in which a potentially releasable product increment is created. However, considering the nature of this project, our team’s sprints will be one week long, starting and ending at each week’s lab. And during each sprint, our members will cooperatively work together to develop one of the many functionalities of the product. To ensure this outcome, we will not make any drastic changes that would endanger our goal and stay focused to deliver what we have planned at the beginning of the sprint. If any problems arise, after each sprint, the scope may be re-clarified or re-negotiated between the Product Owner and the Development Team to ensure all specifications are met appropriately.

In each sprint, a sprint backlog is created from the product backlog, and a shippable product is produced. The checklist for each sprint includes:

* Backlog refinement is completed
* Project tested while integrated with previous sprint product
* Documentation updated if applicable
* Produced shippable product
* Leave incomplete features in the sprint backlog for the next sprint

## The Daily Scrum

Although Scrum requires daily stand-up meetings to share individual progress, due to the unavailability of our members of times and locations, we have decided to cut this feature, and instead hold a daily time-boxed meeting of 15 minutes accessible online via Facebook Messenger to keep in check the development of the project. In this meeting, we will each aim to ask and answer these three main questions:

1. What did you do yesterday?
2. What do you plan to do today?
3. What is preventing you doing these things (if anything)?

## Product reviews

At the conclusion of each sprint, a product review will be performed between the Product Owner and the team to review the progress and status of the product. The new capabilities of the product will be demonstrated, and constructive feedback will be given by the Product Owner, according to the specified requirements.

## Retrospectives

While the Product Review is important to check on the status of the product, a retrospective meeting will also be held for our team to discuss the strengths and weaknesses of the sprint. We will brainstorm ways for the next sprint to become more effective by identifying what went wrong and what could be done better.

## Backlog refinement

In preparation for the next sprint, the Product Owner must get the next few user stories ready. This includes seeking clarification if the user stories are not well understood, and potentially breaking them down into smaller tasks. Existing items in the product backlog are reassessed and discarded if they are no longer relevant or required. Once the product review, retrospective and backlog refinement have been conducted, the team will be ready to start the next sprint.

# Organisation

## Task allocation

For the duration of this project, tasks will be allocated before the start of each sprint, specific to the new capabilities we are aiming to produce by each sprint, according to the sprint backlog. During the meeting in our labs, we will discuss as a team and decide which member will be doing what in order to ensure we can produce high quality results within the timeframe. Our team will be utilising the Trello application (which our tutor has access to) to document each individual user story that must be implemented, in order to keep track of who is assigned to which task. Instead of creating a separate Tasks storyboard, to ensure we know which task is meant for which user story, we will use checklists for each user story within the sprint backlog, which we have implemented. As Trello only allows for a certain number of power-ups, instead of using the Story Points, we will use the checklist feature, and instead leave the Story Points in the description, rather than as a label.

## Time tracking

We will be using Trello also to monitor the time taken for each task by each member. The Trello application includes extensions such as Timelines and Burndown Charts. Instead of using TeamGantt, as we did during our first sprint, we will be using the power-up Chronos Time Tracker. This will allow us to track how much time was spent on each task by each member, instead of the number of days it took. The Burndown Chart power-up: Burndown for Trello will give us an idea of how much work we have left to do, as well as how much work we have done over time. In Agile Tools, there is also a feature where we can add story points to each user story, so we can estimate the relative workload of each task and get a better estimate of how much time it would take. We will utilise these functionalities to optimise our time management for each task and for the overall project. To ensure the data is accurate, at our daily online meetings, we will confirm each team member has correctly and regularly updated the outgoing tasks on Trello.

# Definition of Done

## Checklist for implementing a feature

User Stories describe features that need to be implemented in the project. Implementing a feature will be considered done, if all the following criteria are met:

* Code has been produced for presumed functionalities
* Assumptions of User Story met
* Class diagrams are complete if applicable
* Unit testing is complete, with no errors or bugs
* Feature runs on the intended platforms
* Integration testing is done if applicable
* Proper documentation updated if applicable
* Product owner is satisfied with the feature
* Product performs all important functionalities
* Product tested whether it runs on the intended platform without error
* Systems testing is done
* Usability testing is done
* Client is satisfied with the feature

## 

## 