Project Management Plan

GitHub Inspector

Team Nibble

Ayesha Ali - 29587778

Ryley Angus - 22647597

Shreya Kathuriya - 29506751

Akshith Patil - 29717973

1st September 2019

Last Revised: 29th September 2019

Contents

Summary

The product to be delivered is an inspector for GitHub repositories. It is intended to make the process of marking team projects easier. It achieves this by providing statistics for the individual contribution of each team member. The software will also provide statistics for the entire repository, illustrating information such as amount of lines of code contributed by each team member, the number of issues raised or resolved, and the average time to fix issues. This product will be targeted to the tutors and staff of the Faculty of Information and Technology at Monash University.

Project Vision

To assist Tutors of Faculty of IT in assessing student assignments.

Team Organisation

The team consists of 4 members in their 2nd year of the Bachelor of Software Engineering (Honours) at Monash University (Clayton).

Member Name	User Name	Roles	Responsibilities
Ayesha Ali	aali0016	Designer	- Managing and reviewing the HTML, CSS and User Interface implementation - Ensuring high quality and functionality for the UI design
Shreya Kathuriya	skat0010	Product Owner	- A client representative who will have a vision of the application and will convey that to the team. - Clarifies any doubts in requirements, defines the priority of requirements which go to the Product Backlog
Ryley Angus	rang0004	Scrum Master	- Managing and tracking the progress of the entire project - Ensuring tasks are being completed in a timely manner.
Akshith Patil	apat0017	Tester	- Managing and reviewing documentation such as the Project Management Plan, Risk Register and Product Backlog.

Process Model

As a team, we have decided to follow a model similar to Scrum. This model consists of several sprints. The team will deliver small portions of features in each sprint. For this project, the first sprint was 1 week long and the other two will be 2 weeks long, unlike a typical Scrum process where each sprint is usually 4 weeks long.

After each sprint, a product review with the product owner will be conducted to review the status of the project. The team will also present any new functionality to the product owner. If the project is not yet completed, the remaining tasks will be added to the backlog, in anticipation of the next sprint.

Unlike Scrum, which suggests daily stand-up meetings, we will be conducting online meetings via Discord several times per sprint. These meetings will allow us to update our team members about the progress of the tasks assigned to us and to give feedback.

There are two phases of each sprint. In the first phase, the product owner goes over the highest priority items and in the second phase, the team plans their tasks. In our first phase we have discussed the requirements with the product owner but not task priorities have not been set.

We will be using JavaScript as the primary programming language. The initial code for a feature may also be written in Python, and later converted into Javascript. Other resources which will be used include: Asana, for assigning and tracking tasks per person and time; Git for version control of both code and documentation; Google Drive, to create and share documentation; and Google Sheets for tracking backlogs and burn down charts.

If a particular user story is found to require a substantial amount of investigation and experimentation (a “spike”), the team member responsible for that user story will be expected to document which resources they used, any issues they encountered, and the rationale behind their finished solution. The team member will also be expected to give a brief, high-level overview of their solution at the next team meeting.

Definition of Done

It is a checklist of things which must be completed by the team before the software project is shipped to the client. A user story will not be considered complete unless it meets each requirement on the checklist. It includes:

- Feature complete, bug-free code and all required documentation
- Thorough QA (Quality Assurance) on code and documentation
- Resolution of all issues identified during QA
- Ensuring all requirements and assumptions from user stories are met
- All changes to the code and documentation are recorded
- Approval from all team members
- Unit tests written and passed
- Deployment of software to the required environment

Allocation of Tasks

To ensure maximum productivity and a fair contribution from each team member, tasks will be allocated voluntarily after thorough discussion between team members. Before attempting each task, team members should attempt to anticipate sub-problems within their task and note these in Asana. Any unanticipated challenges or sub-problems should also be described in Asana. This simplifies the process of tracking task assignments, team member contributions and the time spent on each task. Team members are encouraged to consider their workload and level of expertise whilst selecting user stories. This is done to minimise the number of incomplete user stories at the end of each sprint.

Keeping Track of Project Progress

The overall progress of the project will be tracked primarily with Asana. For each task in Asana, deadlines are created and each team member is expected to mark their assigned tasks as being completed upon completion. Asana also shows incomplete tasks, which is useful for tracking progress.

A backlog is a list of tasks which must be done before the completion of the project. It contains a description of the task, an estimation of the effort and time involved and the priority as agreed upon by the product owner and the team. We have decided to use Google Sheets to manage our product backlog.

Tracking Time Spent on Tasks

Time tracking will be managed via Asana. With each task allocation, a deadline for the due date of that task will be set. This will help with time management and allow a reasonably precise task completion time to be recorded. Time will be allocated to each task depending on its complexity and the number of team members assigned to the task. Team members will post the amount of time spent working on their tasks, and exactly what has been achieved, as a comment on the relevant user story in Asana. Team members will be expected to immediately post this information upon the cessation of work.