DevAlarm Deliverable 1:
Project Plan

Team: **Cool Have Fun**

FIT2101 *Software Engineering Process and Management*
01/09/2019

Table of Contents

1. Vision Statement

“For GitHub users who need to keep track of changes to their code contributions and issues assigned, DevAlarm is a web-based notification system that keeps track of changes made to code the user has contributed to or any issues they have been assigned. Unlike the current system where a GitHub user can only be notified of all changes to code regardless of their contribution, our web app provides a personalized notification system.”

---

2. Team Organisation

2.1. Team Members

Team members will communicate via a Facebook Messenger group chat, share assets via a shared Google Drive folder, and communicate via email when necessary.

Name	**Email**	**Roles & Responsibilities**
Ehtesham Ghani	egha0001@student.monash.edu	Frontend developer Backend developer Infrastructure support
Dana Casella	dcas0002@student.monash.edu	Product Owner Frontend developer Backend developer Infrastructure support
Uyen Tran (Sara)	utra0001@student.monash.edu	Scrum Master Frontend developer Backend developer Infrastructure support
Patrick Brett	pbre0003@student.monash.edu	Infrastructure lead Backend lead Frontend developer
Andy Zhan	azha0018@student.monash.edu	Frontend lead Backend developer

2.2. Process Model

The process model for this project is based on the Scrum process framework but with a few adjustments to make it more practical for the team.

Roles

Product Owner

> The Product Owner acts in the interest of the client. Their responsibility is to identify product features, refine and prioritise them accordingly throughout the development process.

Scrum Master

> The Scrum Master's role is to help the team to apply the Scrum process. They are responsible for guiding the team to reach a consensus for a problem and also for minimising outside influence affecting the team.

Team members

> The team is responsible for building the product. The team should be cross-functional and self-managing.

Development process outline

Initializing a product backlog

> A product backlog contains Product Backlog Items (PBIs) - things needed to be done before the end of the project. The product backlog is expected to change over time as the team and client gains more understanding of the product.
>
> PBIs listed in the backlog should have some description, estimated time and priority for completion. The level of detail depends on the item priority, with higher priority items being more detailed.

Sprint planning

> Sprint planning consists of two phases. In the first phase, the product owner explains high priority PBIs to the team and gets feedback from them. The team will then create smaller and detailed tasks based on these items. In the second phase, the team will select which of these tasks are to be implemented in the

sprint and put them into the sprint backlog. These tasks are then assigned to team members.

Sprint

For this project, each sprint will take 2 weeks and there will be 3 sprints in total. During the sprint, the team works on items in the sprint backlog. Unlike conventional Scrum in which the team will have daily Scrum meetings, we will instead have two 3-hour meetings every week to discuss major problems during development. These meetings will consist of a 20-minute Scrum meeting and the rest of the time will be devoted to product development. However, daily updates on progress and problems will still be maintained via messaging on the team Facebook group. With urgent problems, the team will quickly organize a face-to-face or video meeting to resolve them.

Product review and sprint retrospective

Product review and sprint retrospective are done at the end of each sprint, on every second Friday. These dates are the 30th August, 13th September, 27th September, and the 18th October (postponed one week due to mid-semester break). These dates are subject to change upon release of further information.

In product review, the team will do a quick and informal demonstration of the current state of the product. The product owner can give feedback on the new functionalities and refine the product backlog accordingly.

Sprint retrospective is done within the team only. The team will discuss what went well and what went badly in the previous sprint and then refine the process for the next sprint.

3. Time and Task Tracking

The team will use Asana for task allocation, progress tracking and backlog management.

3.1. Allocation of Tasks

Tasks for each sprint will be added on Asana and then assigned to the appropriate team members as discussed during sprint planning. The skills, strengths and desired roles of each individual team member are considered when assigning these tasks, in order to maximise efficiency. Furthermore, a balanced load across the team will be maintained, and any problems with assigned tasks can be discussed during sprint planning and meetings.

Potential sources of problems include silos (in this case, only one or two people in the team having a particular skill set or kind of data). This is especially harmful if a sprint does not require work on a particular section of the project, or requires more work on a particular section than what the only team members with the relevant skills can handle. The solution is to communicate regularly and share knowledge between teammates, incorporating some time for training, and thus achieving a well-rounded team.

3.2. Progress Tracking

The Asana board has five columns:

- Backlog
- To do this sprint
- In progress
- Testing
- Done

The relative progress over the course of a sprint will be determined by the number of items in the ‘testing’ and ‘done’ columns versus those in the ‘to do this sprint’ and ‘in progress’ columns.

3.3. Backlog Management

Product backlog (column name ‘backlog’) and sprint backlog (column name ‘to do this sprint’) items are added to the board in Asana. The items will progress to relevant columns, such as ‘in progress’, ‘testing’ and ‘done’, when they are at the appropriate stage in their lifecycle.

3.4. Time Tracking

When a member starts working on a task, they should move the task to the ‘in progress’ column on Asana. When a task is moved to ‘in progress’, the developer who moved the task should add an empty comment in order to log the time when they started. The due date of a task should also be estimated and input into the task information panel.

The rest of the team should monitor tasks in the ‘in progress’ column and raise concern if a task has taken more than the estimated time and is not finished. Contact the member responsible for that task and help them to resolve any problems.

4. Definition of Done

Our team’s ‘definition of done’ consists of verifying that a task has been completed to a functional standard, having passed automated unit tests as well as human code review and testing. Tasks may still be marked as complete with known issues, and these issues may be resolved in later tasks or sprints.
