Project Management plan

Priyesh Patel – prp6, Matthew Mukalere – mm871, Euel Domingo – ejd29, Sahib Jabbal – ssj9

Communication Methods

WhatsApp Messenger

We will be using WhatsApp Messenger for all our long-distance communication. WhatsApp enables us to send and receive messages to an entire group consisting of all members in the project. We will use WhatsApp's group functionality to create a private group amongst us, in which we will:

- Send each other updates on our individual progress
- Check the progress of other project members
- Ask questions about things we are individually unsure about

Using WhatsApp helps us by allowing us to communicate real-time without the typical delay involved while exchanging emails. The advantages that WhatsApp Messenger provides us over email are:

- Real-time communication (Able to maintain a pace of conversation closer to the pace of Face-to face meetings)
- Faster to use (No need to write individual email recipients or email subject)
- Convenience of use (We all have WhatsApp Messenger installed on our mobiles which we take with us everywhere, and WhatsApp provides us with Push Notifications)

With this real-time communication, we can resolve issues fast, and can return to focusing on more important tasks.

Using WhatsApp also helps us by allowing us to record and take notes on the issues we run into, and need to resolve. So, at the end of every day Priyesh will read the WhatsApp Messenger group conversation to find anything we may be missing from our meeting minutes, such as issues, fixes, and ideas discussed. And since WhatsApp stores all messages we will never miss any key points.

Face-to-face

We will be communicating face-to-face with regular meetings held three times a week, Matthew is in charge of making sure all meetings are scheduled appropriately. On:

- Mondays, at 11am to 12:00am.
- Tuesdays at 12pm to 13:00pm.
- Thursdays at 12pm to 1pm.

Communicating Face-to-face allows us to simultaneously work and communicate at the same time, which will be very good for solving complex problems involving code, or UML diagrams. This also means we will be able to provide each other will real-time help as we will be amongst each other, with laptops and computers. As will be together during face-to-face meetings we will also be able to discuss topics that would take too long to type, allowing us to communicate more and get more done.

We will have roughly an hour to an hour and a half long meetings depending on how long issues take to resolve, if we manage to finish all the tasks set for that meeting, we may end the meeting earlier.

During our face-to-face meetings Priyesh will be taking notes on any key points such as issues, fixes and ideas that we may need to attend to. And at the end of the meeting Priyesh will refine the notes and add them to the meeting minutes

Email

We will be using traditional Emails to communicate back and forth between our client Miles Roman. As this is the only way we can get in touch with our client, we must each be able to send Miles Roman emails in case we have any questions regarding the project. We will contact our client to clarify uncertainties, and for more information on topics which lack sufficient detail. By frequently emailing Miles Roman we will be able to ensure that everything in the development stages through to the final product, precisely fits the clients desired needs.

Iteration planning

To effectively manage each development stage of our project, we will be using the SCRUM methodology. For the SCRUM process to work, we will need to assign a:

- Product Owner - Euel
- Scrum Master - Matthew
- Development Team - Priyesh, Sahib, Matthew, Euel

With these roles appropriately assigned, we will be able to undergo the SCRUM Process. This will involve us gathering user stories and storing them in the Product Backlog, assigning each user story with a Business Value and an Effort rating. With this, Matthew, Priyesh, Euel and Sahib will then begin the Sprint Planning stage, in which we shall break down user stories from the Product Backlog into individual tasks to complete in Sprints, and then store these in a Sprint Backlog. Once we have fully established our Sprint backlog, we can begin the Sprints.

During Sprints, we will participate in Daily Scrum stand-up meetings on WhatsApp Messenger for the sake of convenience. These meetings will be to briefly discuss the progress made from the day prior, and then Matthew will assign everyone tasks for the present day. Having a Daily Scrum meeting will enable Priyesh to monitor the status of our project, detect any issues that need resolving and ensure we are making sufficient progress. The results obtained from these Daily Scrum meetings will be used to update the Sprint Backlog, by marking the status of each task.

While in the process of completing a sprint, Sahib will also make use of a GANTT Chart using Microsoft Project, to visually display the progress of the current sprint according to the Sprint backlog. This will be updated every day based on Priyesh's meeting minutes and progress monitoring, and we should see a steady decline in the amount of work going down. If we do are not making sufficient progress, Matthew will then instruct the team to prioritise tasks, and work smarter in order to get the sprint done on time.

At the end of the Sprint we shall participate in a Sprint Review in which both the Scrum team and stakeholders will congregate to discuss the sprint's outcomes. Euel, The Product Owner will verify whether the resulting product fits the outcomes specified in the user stories, and the Product Backlog will be updated accordingly. The Scrum team will also participate in a separate Sprint Retrospective meeting to analyse the sprint and check for bugs and faults, if any are found, the Sprint backlog will be updated accordingly. The Sprint Review and Retrospection will be held in two of the three weekly face-to-face meetings on a Monday and the following Tuesday respectively.

Group Membership Interchangeability

During our development stages, we will need to ensure that all processes are interchangeable in case one of our group members leaves, requiring us to recruit a new team member. To prepare for such a situation, we will make sure that the tasks required for each role in the group are as transparent as possible, such that each group member knows what everyone else in the group:

- Has done
- Is doing
- Needs to do

This way if one person leaves the group, we can easily replace them and have the new group member up and running immediately, since they will already have tasks to do. This is also an effective way of monitoring each member progress.

Version Control

We will be using GitLab as our primary means of version control throughout this project. To utilise GitLab with ease we will all be using GitLab integration with the IDE IntelliJ. We will also be using GitLab to provide Version Control features to:

- Source code
- UML Diagrams
- Planning Documents

GitLab will allow us to upload code without losing previous versions so in case a rollback is required we can do so easily. This is beneficial because as a collaborative effort, we will all be making changes to the same files separately, and in case one member on the team makes a mistake in a file, we can revert back to an ideal version.

To make sure that our group experience using GitLab is as smooth as possible, we will all be following guidelines to ensure that all updates are as uniform as possible, the guidelines will include:

- Commit messages will need to be descriptive, by containing what changes were made to which files, and what this change resolved. The first line of a commit message must also be longer than 50 characters, this is to increase readability and brevity.

- Version numbering will begin with iterations of 0 for draft versions (0.1,0.2,etc) and with x-n for final versions (1.0,1.1,ect) and when final version amendments are approved x will be incremented (2.0,2.1, ect).

- If merge conflicts occur, the group members involved in the merge conflict will immediately notify each other over WhatsApp Messenger and discuss how exactly to merge the two versions of files together. Doing it this way will ensure that there are no negative repercussions by overwriting important parts of files.

- Pull/Fetch before you push to ensure you have the latest edition of the project in your working tree.

- Commit only when you have completely solved a problem and or have fully working code, to ensure that you do not push "half working" code into the remote repository, which will be susceptible to bugs.

With regards to documents, the footer of all documents shall contain the names of the group members as well as date created / revised where necessary.

Taking all these steps will allow us to work and review efferently and with minimal chance for error (which would have been far greater for not git based control).