Change Management

22/09/2014

Rich Logan

CO600: JustHealth
Supervisor: Yang He

[Version 1.2]
Last saved by: Ben McGregor
Last saved on: 29/03/2015

1.0 Change Management

We have elected to use the source code management software Git in order to control changes, additions and deletions to both our application code and documentation. This will allow all members of the team to add changes without having to worry about conflicts or what the other developers are doing.

Our code will be hosted on the GitHub website in order to provide an easy and visual way of looking at the state of the repository and the log of all commits. The repository is private, ensuring only the developers and can see and interact with it.

1.1 Development Strategy

Using Git, we can develop each of our requirements/features as separate branches of development, ensuring the master branch is not adversely affected by any of those changes. Once that a feature has been completed and has passed all appropriate tests it can be merged back into the master branch, re-running the tests and certifying that feature as complete. The idea is that everything in Master is always deployable at any time.

GitHub also provides issue tracking facilities that we will make use of in order to track issues, bugs and tasks that are in progress. These can be assigned to each team member and also allocated to milestones which will represent each iteration or other stage of development.

Finally, the release functionality will allow us to mark and save the state of the code at the end of each iteration as a deployed and versioned release.

1.2 Development Flow

1. (Milestone created for Iteration)
2. Issue created for specific feature or bug and assigned to milestone if appropriate.
3. Branch created for the issue to avoid disrupting master.
4. Feature/bug is worked on.
5. Issue is closed when ready.
6. Branch is merged to master.