Quality Management Plan

Jensenligan

February 2014

1 Todo

Copy paste these to a valid section, rewrite a bit and if not complete add a slash todo
http://latexforhumans.wordpress.com/2009/03/13/todonotes/).
If you dislike the document structure change it, David
Simply think, what tasks can/do we do to directly improve quality of the product. Or, what are the factors that somehow influence the quality of the product. Everything is valid!!

- Add more todos, increase detail of todos
- How using pivotal improves document-ability and traceability etc.
- How using git improves x.
- How using scrum improves y (see scrum pm).
- How using a (frequent) demo based development improves z (right prioritisation of functionality etc) (see scrum pm)
- Unit tests!!!! A lot.

1.1 Developer interest and productivity

We see developer interest and productivity as our main factor of influence over the quality of the device. The development of the device is done by a small team of 6 members, which means that all members have extremely high influence over the final product. The performance of our team is directly relative to the productivity of each member. It is therefore very important that each member feels connected with the product, its purpose and eager to work with it.

Therefore we

Write about how the fact that we have two identities of the project (formal security device vs toy) increases product quality, how developer interest increases quality and why we prioritize developers over product

2 Code quality

2.1 Modularity and Object oriented design

By focusing on modularity, that is, when logically separable parts of the product is also encapsulated and separated in code. We directly increase the maintainability and flexibility of our code. It gets easier to overview as well as understand and edit. In other languages, encapsulating via classes is the only way to go. However, in python, only separating using multiple files is enough, and this will be our main goal. To try and separate code in as many files possible.

A good object oriented design maximises cohesion and minimizes dependency. Minimal dependency reduces costs of late bug detection.

2.2 Coding standards

By establishing a strict coding standard we make sure that our codebase is easily understood and easily maintained. Our coding standard includes rules such as, name variables, classes and objects according to what they are and

3 Product quality

4 Tasks

Tasks thar will reccur to improve quality Estimated hours....

5 Responsibilities

By working in scrum, all responsibility, including quality assurance's is equally spread over team members. It does however lie in the responsibility of our product owners that the backlog stories they add and prioritize lie in the best interest of the customer.

6 Verification management plan

7 Standard practices

8 Faults management plan

9 Correction costs and early detection

Although a good object oriented design minimizes dependency, all software will contain parts that are transitively dependant on others. This often means that if a bug is created but not discovered, there are risks that continued work of the software becomes dependant on the bug. Hence, the later a bug is discovered, the more dependency and the higher cost.

9.1 Testing

9.2 Problem reporting and corrective action

Bugreport/handling

10 Training

How we will improve our skills, strategies

11 Risk management

Adverse event

12

The device function should be precise with an error margin of +-5 cm from the desired target area.

12.1 Safety

12.2 Usability and ease of use

The user should not be required to have any experience with ssh or Linux to use the final product. (discuss, for example :) For configuration, only editing a config file found on the sd-card is required.