Quality Assurance Document
Version 1.0
CUSTOMIZED POWER EFFICIENT ANDROID ROM

Group Members:
Ajith K.M - ETAKECS004
Anjana Sasikumar - ETAKECS007
K Haridas - ETAKECS026
Under guidance of: Ms. Bisna N. D(Asst.Professor)

April 8, 2013

1 Introduction

Software quality assurance (SQA) consists of a means of monitoring the software engineering processes and methods used to ensure quality. SQA encompasses the entire software development process, which includes processes such as requirements definition, software design, coding, source code control, code reviews, change management, configuration management, testing, release management, and product integration. SQA is organized into goals, commitments, abilities, activities, measurements, and verifications.

1.1 Purpose

Our system aims to improve the power efficiency of the android device by building a customized Android ROM (similar to an operating system) which would make the installation of the stock applications optional. Power efficiency will also be improved by making changes to the kernel to bring about controlling the processor speeds to obtain maximum power efficiency. In building the ROM from the source code additional features like support for native languages like Malayalam can be brought about.

1.2 Scope

Every member in our team is responsible for the actions planned in this document such as documenting the results throughout the development of the project, reviewing the project progress, and testing the project quality, controlled by this plan. The following are the portions of the software lifecycle that are covered by the SQAP: User requirements, Software requirements, Design, Imlementation and Testing.
The list of software items to be covered in each of the above mentioned lifecycle phases are given below:

Software Lifecycle Phase	**Software Item**
User requirements	User requirement document
Software requirement	Software Requirement Specification document
Design	Software Design Document
Implementation	Document including template of functions
Testing	Document showing testing done in project with summary results

The Software Quality Assurance Plan covers the software items. In addition, the SQAP is also covered by this quality assurance plan.

1.3 Reference Documents

- http://en.wikipedia.org/wiki/Software_metric
- http://en.wikipedia.org/wiki/Software_quality

1.4 Project Overview

1.4.1 Background and Context

Our project is to customize an Android Rom thereby enhancing its power efficiency.

1.4.2 Project Objectives

Through this project our aim is to provide a highly customised rom with high performance and at the same time power efficient.

2 Quality Assurance Strategy

To assure quality of software deliverables in each software development phase, we will use the test factor/test phase matrix. The matrix has two elements. Those are the test factor and the test phase. The risks coming from software development and the process for reducing the risks should be addressed by using this strategy. The test factor is that the risk or issue which is needed to be tackled, and the test phase is that the phase of software development which conducts the test. The matrix should be customized and developed for each project. Thus, we will adapt the strategy to our project through four steps.

In the first step, we will select the test factors and rank them. The selected test factors such as reliability, maintainability, portability or etc, will be placed in the matrix according to their ranks.

The second step is for identifying the phases of the development process. The phase should be recorded in the matrix.

The last step is that deciding the test phase of addressing the risks. In this step, we will decide that which risks will be placed each development phase.

Testphase/Test	**Requirements**	**Design**	**Coding**	**Testing factors**
Correctness	Risk: The SRS may not be correct as per the goals of the SQAP Strategy:Formal Tech- view of SRS	Risk: Software design document may not be correct as per the SRS Strategy:Formal Tech- view of SRS	Risk: For lengtheir code functions defined in software design document yearn more for code correctness	Risk User maynot give the correct input.
Performance	Risk: User may have to compromise.	Risk: Software design document may not have the performance as per the requirement.	Risk: Lack of syntatic structures in selected language selected.	Risk: may have several input cases which affect the performance.
Continuity of processing	Risk: Unsatisfiable requirements.	Risk: Improper way of designing requirements	Risk: Possibility of error occurences	Risk: possibility of error occurence.

The matrix forms a part of the quality assurance strategy and as mentioned above this matrix would be used in each of the project lifecycle phases to identify the risks associated in each of the phases with respect to the testing factors. The risks would also be accompanied with their mitigation strategies and in case the risk materialized into a problem, the respective mitigation would be applied. It is for these reasons, that a mention is made about the matrix here in a separate section of the document and not mixed with other sections of the document to avoid repetition.

3 Audits and Reviews

3.1 Work Product Reviews

Work Product	**When reviewed by Quality Assurance**	**How reviewed by Quality Assurance**
Software requirement specification	After a new release	All the requirements raised by the user are achieved. 1. Improved battery efficiency/drain 2. Ability to control the CPU speed of the device 3. Ability to reduce below the default brightness level 4. Removal of bloatwares.
Software document design	After modification	Andoid rom developed. Development done in Java.
Coding	New release	All algorithm and designs coded and implemented perfectly.
Testing	New release	The product is subjected to both blackbox testing and white box testing. All user requirements are met.

4 Further Extension

We have completed our project in all aspects. All functional and non-functional requirements of all four modules are met. The source code was downloaded and compiled in Linux. The product is easily usable and user documentation for all the modules is provided.The system can be extended by adding more features .The project has gone successfully through all the phases of software development and has been tested per the requirements.
Further Extensions include :

1. Adding support to additional languages,specifically Indian Languages.
2. Increased power efficiency.
3. Adding interactive user interface for the installation of applications at the time of ROM installation.
4. Addition of more CPU governors.

5 Conclusion

The Quality Assurance document was developed as per the required standards.