Estima
Quality Assurance Plan

Version <1.0>

Revision History

Date	Version	Description	Author
9 / 01 / 2012	1.0	Initial Quality Assurance Plan	D.A.U.Nanayakkara – 090342F

Table of Contents

Quality Assurance Plan

1. Introduction

The introduction of the Quality Assurance Plan (QA) provides an overview of the entire document. It includes the purpose, scope, definitions, acronyms, abbreviations, references, and overview of this Quality Assurance Plan.

1.1 Purpose

The purpose of this document is to provide a single point of reference on the topic of quality for Estima Project. This document does not contain details review and evaluation techniques, criteria, metrics, etc.

1.2 Scope

QA plan is developed to be used in Estima project. This document describes how product, artifact, and process quality are to be assured using review techniques, audit plans etc. The following stakeholders are the target audience of the document.

- Civil Engineers, Quantity surveyors
- Quality Assurance team
- Project Manager

1.3 Definitions, Acronyms, and Abbreviations

BOQ: Bill of Quantities

1.4 References

Sri Lanka Standard 573: 1999 [UDC 69(083.74)] : Method of Measurement of Building Works (first revision)
Feasibility Study Document
Vision Document
Development case
Software Requirement Specification

1.5 Overview

This document describes organization of development process from the point of achieving the highest software quality. The section *Quality Objectives* describes product quality requirements. The section *Management* describes the organizational structure of quality assurance department, responsibilities and communication between team members in order to facilitate quality assurance. Since the software is one member project all of these roles will be played by one individual. The section *Documentation* gives the list of documents which are designed to check the products' conformity to quality standards. The section *Standards and Guidelines* describes Standards and Guidelines which will be used in product development process. The section *Metrics* defines the metrics that will be measured at certain control points during product development and that will allow controlling the development process. The section *Review and Audit Plan* describes in detail schedule, resources engaged, methods and processes that will be used for project Review and Audit. The section *Tools, Techniques, and Methodologies* describes Tools, Techniques, and Methodologies used in the project. The section *Configuration Management* defines all Configuration and Change Control Management (CCM) activities that need to be performed on the course of project and product lifecycle. It provides the details on the schedule of activities, responsibilities assigned, and resources required, including staff, tools, and computers. The section *Quality Records* describes the process of quality issues tracking. The section *Risk Management* gives the information on how to manage risks associated with the project. This section defines risk management tasks that will be carried out, describes the responsibilities assigned and all additional resources required for the effective risk management.

2. Quality Objectives

Estima must comply with all of the requirements described in Software Requirements Specification (SRS). Product's conformity to Software Requirements Specification (SRS) will be checked through passing the

acceptance tests. Upon client's verification that all of the tests have been satisfactory passed, the product is considered to be of satisfactory quality. It means that product complies with all clients' requirements and is accepted by client.

3. Management

3.1 Organization

Role	Responsibility
Project Manager	Defines priorities, coordinates communication with stakeholders, allocates resources, mostly responsible for keeping the team's focus on main goal, and tries to keep the project team focused on the right goal. He also establishes a set of practices that ensures integrity and quality of project artifacts.
Software architect	Designs the overall system architecture and decomposition of overall architecture into subcomponents. Lead and coordinate technical activities throughout the project.
System analyst	Do requirement elicitation using RUP tools like use-case modeling, context models. Categorise system requirements into functional requirements, non-functional requirements. Prioritizing requirements.
Developer	Responsible for developing and testing components as per the standards of the Estima system. The developer shall adhere to design principles and coding conventions of the system.
Test Designer	Responsible for planning, design, implementation, and evaluation of the tests, including: ▪ Developing the test plan and test model ▪ Implementing the test procedures ▪ Evaluating test coverage, test results, and test effectiveness ▪ Generating the test evaluation summary
Tester	The Tester is responsible for test execution, including test set-up and test run, evaluation of test run and error recovery, defect logging and test results recording.

Since the initial phase of the project is done by one person (D.A.U.Nanayakkara) all the roles are played by him. After it becomes a open source project this may change

3.2 Tasks and Responsibilities

To assure the high quality of "Estima" system the following actions must be done:

- Project Review and Audit according to Review and Audit Plan (see section 7). Responsibilities are described in corresponding section of this document.
- Software Testing according to Test Plan (see section 7). Responsibilities are described in corresponding section of this document.
- Acceptance testing according to Acceptance Test Description. Responsibilities are described in corresponding section of this document.

4. Documentation

Feasibility study document contains information gathered to test the feasibility of the system. This contains competitors of the product and what differentiates the product from the rest and the problem that this software tries to cater.

Project Vision Document elaborate on the problem that the software system is going to cater for and the basic feature needs to be implemented. This gives an idea of the range in which this software system operate and the main goals of the project relative to its users and client.

Project Development Case gives an in depth explanation on the development procedures used in the project. This document describes ways in which the RUP will be adapted for Estima project.

Software Requirement Specification fully describes the external behavior off the Software system, "Estima". This document describes the functional requirements, nonfunctional requirements, design constraints and other factors necessary to provide a complete and comprehensive description of the requirements for the system.

Software Architecture Document provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system.

5. Standards and Guidelines

Estima system shall be developed according to the following standards and guidelines:

- Presentation of B**OQs** (format and nomenclature) shall adhere to SLS 573: Method of Measurement of Building Works (first revision).
- The user interface of the system shall confirm to the guidelines provided by Qt framework. http://doc.qt.nokia.com/qtcreator-extending/qtcreator-ui-text.html
- Whenever possible interfaces shall use domain specific terms. (to make the domain experts comfortable in using the software)
- Use-cases and architecture design shall adhere to RUP framework tools provided.

6. Metrics

Metric	Purpose	Sample measures/perspectives
Progress	Iteration planning Completeness	No. of functional requirements completed
Quality	Iteration planning Rework indicator Release criterion	Number of errors Defect discovery rate Defect density
Maturity	Test coverage/adequacy Robustness for use	Test hours/failure and type of failure

7. Review and Audit Plan

Review and Audit Tasks

- Requirements review is done after completion of each requirement under iterations. The review is done as per the Software Requirement Specification (SRS)
- Architecture Review is performed at the end of elaboration phase. And the compliance of software to the architecture is done throughout the project whenever new feature is added to the system. Software Architecture Document is in this scenario. Architecture review is done by the project architect.
- Code review is done to check the source code. This will include testing source code for errors in coding conventions used, performance wise issues etc.

Organization and Responsibilities

- Requirements Review

  Project technical assistant or Review intendant is responsible for the initial creation of a review report. He analyzes Software Requirement Specification and presents the results of

his work to development team. He is also responsible for revision history keeping. Software architect and Software analyst are responsible for updating and making corrections of Requirements review. They are also responsible for the revision history keeping.

- Architecture Review

  Project technical assistant or Review intendant is responsible for the creation of Architecture review, which is devoted to software Architecture analysis He is also responsible for the revision history keeping. Software architect and Software analyst are responsible for updating and making corrections of Requirements review. They are also responsible for the revision history keeping.

- Code Review

  Developer or Review intendant is responsible for creation of the initial review as a text-file kept in the Source Control System. Code Review is a result of source code proof reading and checking its compliance with Code Requirements.

8. Tools, Techniques, and Methodologies

Techniques

- Black Box Testing

  Black-box tools deal only with use cases or functional description of the test target compared with white-box tools, which are based on the knowledge on how the test target processes the request. Black-box tools rely upon the input and output conditions to evaluate the test.

- White Box Tests

  White-box tools rely upon the knowledge of code, design model(s), or other source material to implement and execute the tests.

- Ad Hoc Testing

  Ad hoc testing involves the use of unstructured or previously unplanned activities. This testing method provides the test methodology with a degree of variability.

9. Configuration Management

Throughout the project all source files and documentation are tracked through a revision control system. The system that is used is GIT. When a significant change occur in the system or a bug is fixed the system is updated using the revision control.

10. Quality Records

All defects are tracked through a internal document that is used to track defects. Each defect will be categorized according to whether the defect is corrected or not. And the defects that are not corrected are ranked according to the risk each defect has on the system.

11. Risk Management

Risk Management is done by making a risk list and it is supported during the whole project life cycle. The risks are prioritized according to the impact the risk has on the system. The person responsible for creation and support of the Risk List is Project Manager. All team members take part in creation and maintenance of the Risk List, making necessary marks in its revision history.