Software Quality Assurance Plan: MSE Portfolio Project

October 24, 2008

Prepared by Doug Smith
Version 1.0

Table of Contents

Revision History

Version	Date	Changes
1.0	10/23/2008	First draft.

10/29/2008 8:57 PM

Purpose

The purpose of this document is to define a software quality assurance (SQA) plan for the MSE Portfolio project.

Reference Documents

All documents are available on the project website associated with this effort.

[1] Vision Document: MSE Portfolio

[2] Project Plan: MSE Portfolio Project

Management

Organization

This organizational structure of this project consists of the following individuals:
- Doug Smith: Student
- Dr. Daniel Andresen, Major Professor and Committee Chair
- Dr. Mitchell L. Neilsen, Committee member
- Dr. Gurdip Singh, Committee member

During the elaboration phase of the project, at least two other MSE students will participate as reviewers in this project. The specific individuals are TBD.

Tasks and Responsibilities

The SQA plan covers the entire life-cycle of the project.

For this project, the student is responsible for the creation and quality assurance of the artifacts according to the project plan and in alignment with the standards and expectations of the MSE Portfolio Project, and the MSE program in general, and for recording, acting on, and demonstrating compliance to any suggested actions by the committee.

The committee is responsible for reviewing the artifacts at the checkpoint presentations at the end of each phase, and for ensuring the work is performed to the satisfaction of the committee and in accordance to the expectations and standards of the MSE program.

Documentation

The documentation artifacts for this project are outlined in the MSE Portfolio requirements, specified at this location: http://mse.cis.ksu.edu/documents/MSE-portfolio.pdf

The documents are reviewed by the project committee at the end of each phase. The specification of the required artifacts details the phase at which artifacts are produced, updated, and reviewed.

All artifacts will be stored on the project website.

Standards, practices, conventions, and metrics

Standards

The standards governing this project are:
- Documentation standards – MSE Portfolio
- IEEE Standard for Software Test Documentation (IEEE Std 829-1998)
- IEEE Standard for Software Quality Assurance Plans (IEEE Std 730-1998)

Metrics

The following metrics will be recorded as part of this project:
- Source lines of code produced by the project
- Time spent during the project

Reviews and audits

The MSE Program includes 3 audit reviews, with a review occurring at the of each phase in the software development life-cycle described in the project plan. There will also be an architectural design inspection performed during the elaboration phase of the project.

Test

All tests associated with this project will be documented in a test plan prepared during the elaboration phase of this project.

Problem reporting and corrective action

Formal problem reporting occurs at the ends of presentations I and II. The problems will be recorded in an action document, and corrective action associated with the action document will be review in the following presentation.

Tools, techniques, and methodologies

In general, test-driven design will be used to do unit testing, with JUnit being used as the unit test framework. Integration testing will be done using SoapUI, and scale testing will be done using either SoapUI or JMeter.

Code control

A configuration management system (either Clearcase or Subversion) will be used to control the source code artifacts produced as part of the project. Document revisions will be maintained in a sharepoint repository, with release versions of documents made available individually on the project website.

Media control

Appropriate backups of source code and artifacts are the responsibility of the student.

Supplier control

This section is not applicable to this project.

Records collection, maintenance, and retention

Test results (especially performance and scale test results) produced by this project will be captured and made available on the project web site.

Training

No training specific to this project is required.

Risk management

In general, risk is managed in this project via the inception phase architectural proof of concept demonstration, and in the production of the executable architectural prototype produced in the elaboration phase. This confronts problems in the execution of the project early enough to allow the project to be re-scoped as appropriate.