E-Store

Software Quality Assurance Plan

version 0.2.0, 20 Feb 2016

Project Team:	Roselle Ebarle Catherine Maglasang Dexter Esin Mar Yee
Project Manager:	Roselle Ebarle
Senior Manager:	Catherine Maglasang
Advisor:	Orven Llantos
Customer:	

Abstract

This document is about the Software Quality Assurance Plan (SQAP) of our system, e-store which is an open-source, python-based ecommerce platform for online merchants. It provides the power to grow your web business, reach more customers and sell more products and services. It enables businesses to experience an integrated workflow for their business – Sales, Inventory & Order Management and Customer Service under one platform.

Contents

Document Status Sheet

Document Title	Software Quality Assurance Plan
Document Identification	E-Store
Author(s)	Ebarle, Maglasang, Yee, Esin
Version	0.0.1
Document Status	Draft

Version	Date	Author(s)	Summary
0.0.0	05-02-2016	R. Ebarle, C, Maglasang, M. Yee, D. Esin	Document Start
0.1.0	13-02-2016	R. Ebarle, C, Maglasang, M. Yee, D. Esin	Edited Chapter 1, Started Chapter 2
0.2.0	20-02-2016	R. Ebarle, C, Maglasang, M. Yee, D. Esin	Edited Chapters 1 and 2, Started Chapters 3 and 4
0.3.0	10-03-2016	R. Ebarle, C, Maglasang, M. Yee, D. Esin	Started Chapter 5
0.4.0	11-03-2016	R. Ebarle, C, Maglasang, M. Yee, D. Esin	Edited Chapters 4 and 5, Started Chapter 6

Chapter 1

Introduction

1.1 Purpose

This document describes the procedures and control methods to obtain the desired quality level of the end product and the process by which the end products are created. This document serves as a guide for the managers and developers of the e-store project. All team members must read this document and apply the procedures stated in it.

1.2 Scope

The e-store project should follow the RESTful architectural style. The entire system has to be developed using the Flask microframework, with ajax doing the client requests. There will be a clear separation of concerns between the client and the server for easy maintenance and scalability.

1.3 List of Definitions

ATDD	Acceptance Test Driven Development
TDD	Test Driven Development
BDD	Behavior-Driven Development
SQA	Software Quality Assurance
UML	Unified Modeling Language
ERD	Entity Relationship Diagram
MSU-IIT	Mindanao State University - Iligan Institute of Technology
REST	Representational State Transfer
FLASK	Web Framework
Python	Programming Language
SCS	School of Computer Studies
Sales Inventory	The list of items such as the goods that are in stock
E-commerce	The buying and selling of goods over an electronic network, primarily the internet
Customer	The person who transacts in the store page of the business
Admin	The owner of the products sold in the store.
Product	The items being sold in the website

Cart	The list of items the customer is going to buy
Checkout	The process in which the customer is going to buy and pay the items inside the cart
Gherkin	Business Readable, Domain Specific Language that lets you describe software's behaviour without detailing how that behaviour is implemented.
QAM	Quality Assurance Manager
SQAP	Software Quality Assurance Plan
SQMP	Software Quality Management Plan
PM	Project Manager
CM	Configuration Manager
AD	Architectural Design
DD	Detailed Design
CI	Configuration Items
UML	Unified Modeling Language
SVVP	Software Validation and Verification Plan
Lettuce	Framework for Behavior Driven Development in Python language

1.4 List of references

[SQAP] Software Quality Assurance Plan, SPINGRID team, TU/e, 0.1.3, June 2006
Saleh H. (2013). *Javascript Unit Testing*. Packt Publishing
Osmani A., (2012). *Javascript Learning Design Patterns*
Zlobin G., (2013). *Learning Python Design Patterns*
Sale D., (2014). *Testing Python*
IEEE Standard for Software Quality Assurance Processes, IEEE Std 730-2014
Clean Code Cheat Sheet
Test-Driven Development, Dr. Christoph Steindl, Senior IT Architect and Method Exponent, Certified ScrumMaster
Best Practices, Development Methodologies, and the Zen of Python, Valentin Haenel
Test-Driven Development, Gary Brown
Detailed Design, (2006). Parametric Technology Corporation (PTC)
Configuration Items, http://www.chambers.com.au/glossary/configuration_item.php
Software Validation and Verification Plan, https://cours.etsmtl.ca/mgl800/private/Normes/ieee/guide%20for%20software%20verification%20and%20validation%20plans.pdf
Lettuce Framework, http://c2.com/cgi/wiki?LettuceFramework

Chapter 2

Management

This chapter details the structure and tasks of the Software Quality Assurance (SQA) team.

2.1 Organization

For a survey of 40 the organization within the project and the responsibilities of the individual members of the teams see [SPMP]. The SQA team is led by the Quality Assurance Manager (QAM). The vice-QAM assists the QAM. Furthermore, the QAM is responsible for the SQA and the performance of the SQA team. The QAM will contact the PM if the software quality is endangered.

2.2 Tasks

The SQA team’s main task is to check whether the procedures are followed and that standards are handled correctly as defined in the

[SQAP], [ATDD], [BDD], [TDD], [REST], and Gherkin. Additionally, the SQA team inspects whether all group members fulfill their tasks as defined in [SPMP] according to the parts of the [SQAP] applying to their specific tasks. Besides the described main task, the SQA team has the following additional tasks:

- The SQA team has to check the consistency and coherence between documents.

‘

2.3 Responsibilities

As described in section 2.2, the main responsibility for the SQA tasks lies with the QAM. Within the SQA team, the QAM can delegate the tasks. Minor problems can be solved by each member of the SQA team, whereas major problems are matter of the QAM and are also reported to the PM. Every problem found by a team member has to be reported to the QAM. Reporting to project members outside the SQA team is done by the QAM. The vice-QAM will assume his tasks if the QAM will be unavailable for a short period of time. If the QAM will be unavailable for a longer period of time, the SQA team must be expanded and the tasks reorganized. The project manager will be responsible for this.

‘Chapter 3

Documentation

The documents to be delivered in the specific phases of the project will be based in Chapter 4. Document standards are described in the same chapter.

Chapter 4

‘

Standards, Practices, Conventions And Metrics

4.1 Documentation Standards

During this project many different documents will be made. Every document has to be approved by:

- The author(s)
- The leader of the responsible team
- A member of the SQA team

In case that these three people happen to be one and the same, a second member of the responsible team has to give these approval as well. Only approved documents affect the project. The documentation standards involve the following:

- All documents must be written in English.
- Requirements on review and approval as described in Chapter 5.
- Procedures involving the change of documents.

These standards apply to all documents, to electronic versions as well as printed ones. However the layout requirements do not apply to documents other than the project and product documents. All documents made are available through the document repository. In case of unavailability of the document repository, the CM sees to it that there are three copies available of every document (latest version with

the highest status of approval) in the group's workspace. The three copies consist of one copy on paper and two digital copies on two different geographical locations.

4.2 Design Standards

The design standards in the Architectural Design (AD) and Detailed Design (DD) phase will be defined. The software design paradigm that will be used is Object Oriented Programming. UML will be used as modeling technique for object oriented designs.

4.3 Coding Standards

Coding standards will be based in the Python Coding Standards.

4.4 Comment Standards

Comment standards, they form a part of the coding standards. Thus, it will be described based in the Python Coding Standards.

4.5 Testing Standards

Testing standards will be based according to [BDD].

4.6 Metrics

Members of the SQA team will measure the delivered software's quality during random checks by means of metric. Some examples of metric are:

- Length of procedures (should not be more than 100 lines)
- Total length of (useful) commentary divided by the total length of code (should exceed 1/50)
- Number of parameters divided by number of procedures (should not exceed 5. Procedures should have no more than 7 parameters)
- Nested if-statements' maximum depth (should be less than 4)
- Nested loops' maximum depth (should be less than 3)

The SQA team makes a small report containing the results of the described tests in its log and delivers the report to the author(s) of the checked document. If a violation of these metrics is detected, it has to be resolved, unless the PM and QAM grant a permit.

4.7 Compliance Monitoring

The SQA team will monitor compliance to the proposed conventions by the way of random checking of Configuration Items (CI) during which references to other documents are checked. During the reviews the SQA team member present checks with the authors of the reviewed CI whether it has references in it. He/she will also check whether the authors checked CIs that was referenced. Discovered problems are reported to the PM.

Chapter 5

‘

Review

Standards and procedures for reviews and audits will be based according to [SVVP]. With respect to it, the SQA team checks before the internal reviews whether it contains:

- reviewing and audits;
- testing;
- tracing;
- the Acceptance Test Plan;
- the System Test Plan; and
- the Integration Test (IT) Plan;

In addition to this, the SQA team carries out random checks as described in the next paragraph.

The SQA team randomly checks all project and product documents, and the program code whether it followed the document and code standards respectively, and all members of the group do their tasks well. Random checks are an addition to the reviews. Every document and code undergo a random check at least once. In order to save time, the SQA team doesn’t need to write a report. During a progress meeting (possibly), it just reports the results to the Project Manager and Senior Manager. If problems are detected, a date is set for the problem that must be solved and then the document and code are checked again.

Chapter 6

‘

Test

Methods and procedures for testing are detailed according to a framework called Lettuce. In random tests, from Chapter 5, the SQA team observes that these procedures are followed. If the testing procedures are not followed, the PM will be informed by the SQA team.