

Cafeteria Management System (Cashless Canteen)

Quality Assurance Plan 2.0

Team-2

Instructor – Professor Asim Banerjee

25th October, 2013

DA-IICT, Gandhinagar

Overview:

This Software Quality Assurance Plan (SQAP) document describes various quality measures for the Cafeteria Management System.

Target Audience:

- Students
- Faculty
- Staff
- Software development team

Document Revision History:

| Version | Primary Author(s) | Description | Reviewed By | Date |
|---------|-------------------|-------------|-------------|--------------|
| 1.0 | Ankush | SQAP v1.0 | Dipen | 25 Oct, 2013 |
| | | | | |
| | | | | |

Table of Contents

| | |
|--|----------|
| 1 Introduction | 1 |
| 1.1 Purpose | 2 |
| 1.2 Scope | 2 |
| 2. Management | 4 |
| 2.1 Organization | 2 |
| 2.2 Scope | 2 |
| 3. Software Quality | 4 |
| 4. Documentation..... | 4 |
| 4.1 Purpose | 2 |
| 4.2 Documentation Requirements | 2 |
| 5. Standards & Conventions..... | 4 |
| 5.1 Documentation Conventions | 2 |
| 5.2 Coding Conventions | 2 |
| 6. Software Quality Assurance Activities..... | 4 |
| 6.1 Software Quality Control | 2 |
| 6.2 Inspection & Review | 2 |
| 6.3 Audits | 2 |
| 6.4 Verification & Validation | 2 |
| 6.5 Testing | 2 |
| 7. References | 4 |

1. Introduction

1.1 Purpose

Software Quality Assurance Plan (SQAP) consists of those procedures, techniques and tools used to ensure that a product meets the requirements specified in software requirements specification.

SQA encompasses a quality management approach, effective software engineering technology – methods and tools, formal technical reviews that are applied throughout the software process, a multi-tiered testing strategy, control of software documentation and the changes made to it, a procedure to ensure compliance with software development standards, and measurement and reporting mechanisms.

1.2 Scope

The scope of this document is to outline all procedures, techniques and tools to be used for quality assurance of this project.

The use of this plan will help assure the following:

- ☐ To establish a quality management approach
- ☐ Identify the SQA responsibilities of the project developers and the SQA consultants
- ☐ Identify the SQA work products
- ☐ The software development, evaluation and acceptance standards are developed, documented and followed.
- ☐ The results of discussions in various review meetings conducted are reflected in the corresponding deliverables.
- ☐ The test results adhere to the acceptance standards.

2. Management

2.1 Organization

We've planned that the products at the end of each phase in the lifecycle of the software would be reviewed by the team members, and the changes suggested would be incorporated by the corresponding reviewer and the author of that document. In case of the code segment which is being reviewed, the changes would be incorporated by the author of that code segment, based on the suggestions by the reviewer in the meeting.

2.2 Roles and Responsibilities

Different members of the entire team are assigned a role responsible for quality management of the project. In this way entire project development team plays significant role in SQA.

- ☐ Project Manager: Purav Chitalia

The Project Manager is responsible for the implementation of the SQA as well as ensuring all products, processes, and documentation adheres to the plan.

- ☐ Project Team

Project team members – module managers, designers, developers and testers, participate in the quality assurance and control process and follow the SQAP to bring forth a good quality product, which works best for the client in minimal cost.

3. Software Quality

Structural Quality: It is evaluated through the analysis of the software's inner structure, its source code, in effect how its architecture adheres to sound principles of software architecture. So, the structural quality of our software should reflect how well it complies with or conforms to a given design, based on functional requirements or specifications.

Functional Quality: It is typically enforced and measured through software testing. Refers to how it meets non-functional requirements that support the delivery of the functional requirements, such as robustness or maintainability, the degree to which the software was produced correctly.

Certain activities must be performed to ensure the structural and functional, i.e. the overall quality of the product, namely:

- ☐ Maintenance of documentation standard. All documents must be in proper format, technically accurate and completed on time.
- ☐ Inspection and reviews of all the documents. Every document is reviewed by some other team member and changes are suggested. These changes are then incorporated according to the configuration management plans.
- ☐ Coding conventions are followed in the whole project coding and implementation phase. This makes the software code easy to understand and to maintain.
- ☐ Verification and validation activity is done at the end of every phase. It checks where the present phase is complemented by the assumptions made in the previous phase.
- ☐ Easy to maintain. This makes a software adaptable to the constantly changing needs of the customer, the hardware etc.

- ☐ Verify that relevant documents are updated and based on approved requirements changes.
- ☐ Configuration management plan is properly followed to verify change control integrity.
- ☐ Quality assurance is followed through the whole Project Life Cycle.

4. Documentation

4.1 Purpose

This section will identify the documentation governing the development, verification, validation, use and maintenance of the software. Evaluations of specified documents to determine conformance to applicable standard, as well as to ensure format, editorial, and technical adequacy is done. Document evaluation results will provide corrective actions for non-conformance to requirements.

4.2 Documentation Requirements

Following documents are integral part of any project and must be made so as to ensure that every Software Engineering philosophy is followed and that quality of the product is ensured. The following documentation will be generated and updated throughout the duration of software life cycle:

- A. Project Plan:** Purpose of this document is to give a complete yet concise description of purpose, goal, different criteria, phases and timings of each phase, monitoring and guidance, risk, installation, cost of the project. It serves as a guide for the process of software development. While during the development of the software, the model used for description and the norms specified in this document should be followed so as to ensure that the software is a quality product.
- B. Software Requirements Specifications:** This document gives a holistic overview of the Software Requirements for the software. It is the official statement of what the system developers should implement and it helps to provide an organized way to collect all requirements surrounding a software intensive project at the feature level into one document.

SRS document should clearly specify:

- ☐ Functional requirements of the system. It should specify input, transformation, and output requirements of each function.
- ☐ Non-functional requirements of the system e.g. Usability, Maintainability, Portability, Reliability, Accuracy, Human-computer interface issues, Performance.
- ☐ Constraint under which system must work e.g. co-existence with other systems.

To ensure that the software is a quality product, SRS should be thoroughly revised and reviewed. It must be ensured that all features of the product that are mentioned in SRS are incorporated and met with desired quality in the final product

C. Design – High Level & Low Level Document: After the purpose and specifications of software have been defined, we need to do the basic designing of the software. This software design document presents the designs used or intended to be used in implementing the software product. It gives a detailed description of the design aspects of the software i.e. User Interface, Database, Implementation Architecture Style and Procedural design. The design document accompanies an architecture diagram with pointers to detailed feature specifications of smaller pieces of the design. Practically, a design document is required to coordinate a large team under a single vision. It describes the architectural style and the Architecture followed to develop the software. The design document should be reviewed thoroughly to ensure that it is in perfect sync with the SRS, gives a clear perspective to the coders to implement the design, and thus is critical to develop a good quality software product.

D. Risk Management & Mitigation Plan: This document describes the risks that the team may encounter while designing and managing the project and the strategies and actions that the team will conduct to mitigate these risks. Risk management means dealing with a concern before it becomes a crisis. It is a software engineering practice deal with processes, methods, and tools for managing risks. It is a continuous and developing process that runs through the project strategy and the implementation of that strategy. It addresses methodically all the risks surrounding the project activist past, present and in particular future. It increases the probability of success, and reduces both the probability of failure and the uncertainty of achieving the overall objectives.

E. User Manual: This document shall specify and describe the user interface, required data and control inputs, input sequences, options, program limitations, and other activities or items necessary for successful execution of the software. All error

messages shall be identified and corrective actions shall be described. A method of describing user identified errors or problems to the developer or the owner of the software shall be described. A detailed and accurate user manual would help the users in understanding and operating the software product smoothly, thus adding to the overall quality of the developed software.

F. Software Configuration Management Plan: The SCMP shall document methods to be used for identifying software items, controlling and implementing changes, and recording and reporting change implementation status.

5. Standards & Conventions

5.1 Documentation Conventions

Documentation standards have been developed with references to IEEE standards and previous year Software Engineering project documents. And the work product for the same has been generated.

All the documentation reviews have strictly followed the developed standard to ensure technical and editorial clarity and accuracy of the documents.

5.2 Coding Conventions

Standard coding conventions for HTML, CSS, JavaScript and Java Web Development were followed. The work product for the same is generated and is followed for writing code.

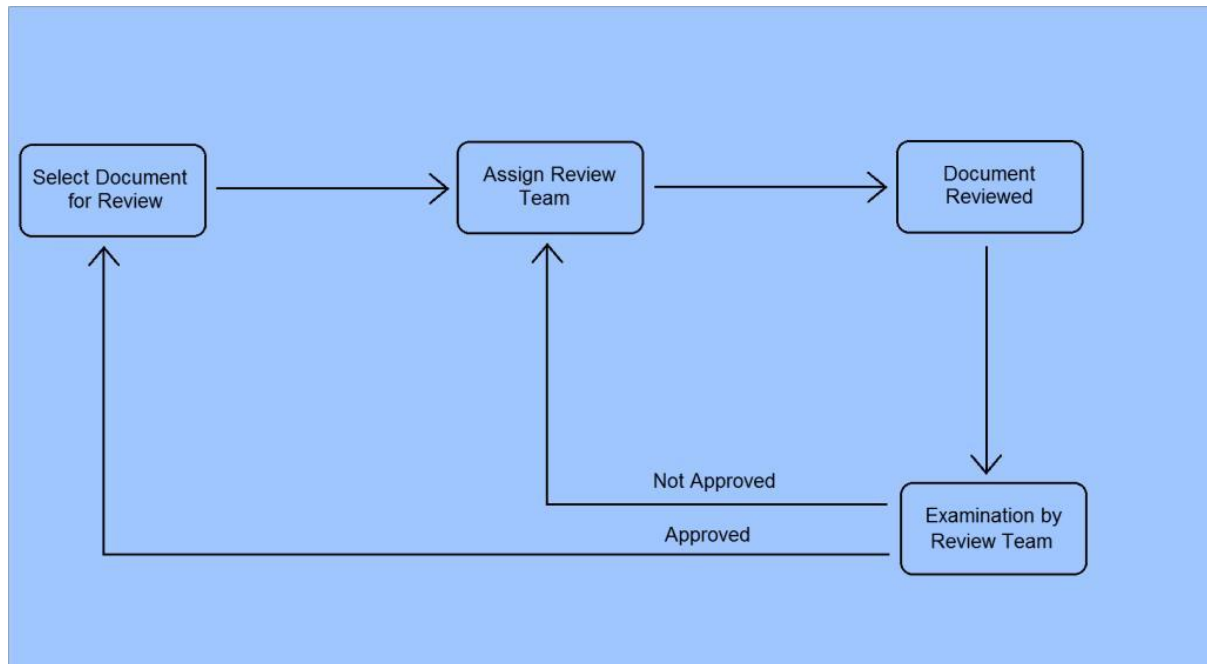
6. Software Quality Assurance Activities

6.1 Quality Control

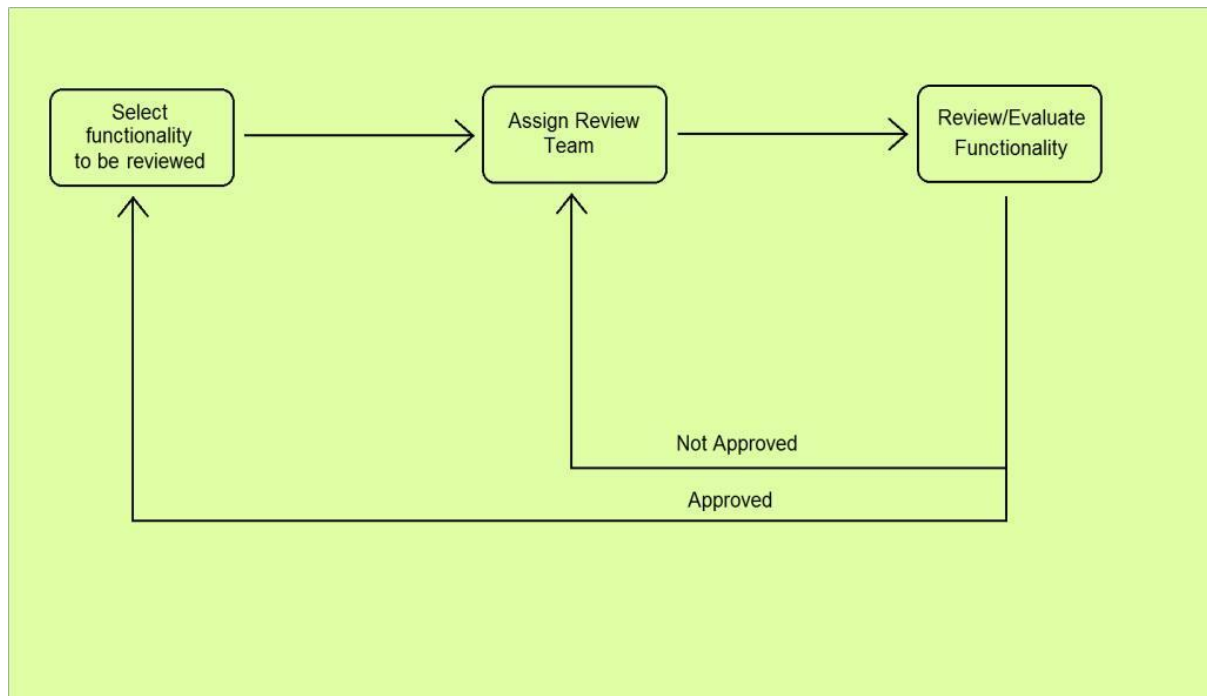
When a document is selected for review, a review team is assigned to that document which performs a quality review of the document. The factors a document is checked for are: - completeness, correctness, easy to understand and ensuring there is no ambiguity.

After review, the team then undertakes an examination of the document to discover any problems. If the document is approved, then another document that needs reviewing is selected. If the document does not get an approval, then it is again assigned a new review team which carries on with the review process as shown above.

Quality Control for Documents:



Quality Control for Software:



6.2 Inspection & Review

Before the document is drafted by the drafting team, a template is adopted in discussion with the Project Manager. After the first draft of the document is prepared, the document under consideration is read thoroughly by the member assigned to the task. He/she checks whether the document satisfies the norms that have been set down or rather the template that has been adopted for that particular document. In case any anomalies are identified, the reviewer lists them in a document.

An anomaly as per IEEE standards is: Any condition that deviates from expectations based on requirements specifications, design documents, user documents, standards, and so on or from someone's perceptions or experiences. Anomalies may be found during, but not limited to, the review, test, analysis, compilation, or use of software products or applicable documentation. Then in the next meeting the anomalies are discussed and changes are generated. Thereafter, the document is changed and the changes are again recorded in the review document.

The norms that need to be followed during a review are:

- ☐ Each section should be relevant to the matter at hand.
- ☐ The content of a section should be simple, lucid and elegant.
- ☐ No details pertaining to the phase or sub phase that is being documented should be left out.

6.3 Audits

Regular audits of the SQA activities are held. These audits allow the member to determine which SQA activities are working to deter product defects and assess how well each SQA activity is being conducted. The audits assist in keeping the SQA teams on track, as well as enhance beneficial SQA activities or eliminate useless activities. Each SQA activity is assessed to determine how well it is affecting product quality.

Team members will have a report on their individual performance in every meeting. Any problems, question regardless on the performance of other team members will be also noted there.

- Any changes that will affect the project will be presented to other team members before doing any changes. These are the changes that are minor or require little code change or change in any documentation.
- The client should be notified of all changes made. For minor changes, we will just notify a representative from the client instead of the whole team from the client.
- This rule only applies to the minor changes or cosmetic changes, or minor functional changes. Any major functional change will still require the agreement from the client side.
- A changed version will also be carefully recorded. Detailed review log mentioning what kind of change will be making on where and by who will be created.

6.4 Verification & Validation

Verification: It is the process of determining whether or not the products of a given stage of the software development cycle fulfill the requirements established during the previous stage.

The following activities will be performed as part of requirements verification:

- ☐ Evaluate SRS requirements and relationships for correctness, consistency, completeness, accuracy, readability and testability.
- ☐ Produce a traceability matrix tracing all requirements.
- ☐ Assess the criticality of requirements to identify key performance or critical areas of software.

Validation: It is the process of evaluating software at the end of the software development process (acceptance testing activity in the Testing stage) to ensure compliance with software requirements.

The following activities will be performed as part of validation:

- ☐ Plan acceptance testing, including criteria for:
 - o Compliance with all requirements.
 - o Adequacy of user documentation.
 - o Performance at boundaries and under stress conditions.
- ☐ Plan documentation of test tasks and results.
- ☐ Execute the Acceptance Test Plan.
- ☐ Document acceptance test results.

6.5 Testing

Software testing is any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results. It detects the differences between existing and required conditions and to evaluate the features of the software item. The purpose of testing can be quality assurance, verification and validation, or reliability estimation.

The Test Plan document is designed to describe the scope, approach, resources and schedule of all testing activities. The plan identifies the items to be tested, the features to be tested and the type of testing to be performed. The main purpose is to ensure that all the modules are tested according to the specifications described in the SRS and that all the modules are working properly.

At the end of test phase, the software should fit the metrics for each quality requirement and should satisfy the use case scenarios and maintain the quality of the product. At the end of the project development cycle, the developer/owner of this application should find that the project has met or exceeded all of their expectations as detailed in the requirements.

Another objective of System testing is to identify and expose all issues and associated risks, communicate all known issues to the project team, and ensure that all issues are addressed in an appropriate matter before release. Any changes, additions, or deletions to the requirements document, Functional Specification, or Design Specification will be documented and tested at the highest level of quality allowed, within the remaining time of the project and within the ability of the test team.

...

7. References

IEEE Std 730-1998 (Revision of IEEE Std 730-1989) – IEEE Standard for Software Quality Assurance Plans