

E-Gram Seva

Quality Assurance Plan v1.0

Team 22

April 1, 2013

REVISION HISTORY

Version	Author	Date
Version 1	Surbhi Singhal	April 1, 2013
Reviewed	Rutvik Jhala	April 8, 2013

CONTENTS

1. Introduction.....	4
2. Purpose.....	4
3. Acronyms.....	4
4. Reference Documents.....	4
5. Tools, Techniques and Methodologies.....	4
6. Software Quality Management Plan.....	6
6.1 Quality Planning.....	6
6.2 Quality Assurance Plan and Quality Control.....	8
7. Software Documentation.....	9
8. Review Reports.....	10
9. Validation and Verification.....	11
10. Communication and Reporting.....	11

1. Introduction

Software Quality Assurance (SQA) is a formal process for evaluating and documenting the quality of the work products produced during each stage of the software development lifecycle. It aims at ensuring high quality software according to stated requirements and standards.

2. Purpose

The purpose of Quality Assurance Plan is to set goals, responsibilities and appropriate processes which are required to assure Quality for the project. The Quality Assurance Plan defines techniques, methodologies that assures quality, timely completion and deployment of the software.

3. Acronyms

- SCM : Software Configuration Management
- SCMP : Software Configuration Management Plan
- SRS : System Requirements Specification
- SDS: System Design Specification
- QC: Quality Control
- SQAP: Software Quality Assurance Plan
- PDCA: Plan-Do-Check-Act

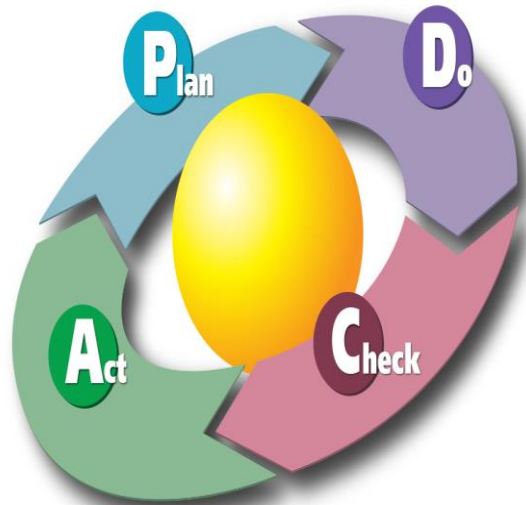
4. Reference Documents

- Project Plan
- SRS

- Design Document

5. Tools, Techniques and Methodologies

The team's effectiveness and efficiency in achieving its quality objectives are contributed by identifying, understanding and managing all interrelated processes as a system. Our team is inspired by Deming's PDCA Cycle.



PLAN

Establish the objectives and processes necessary to deliver results in accordance with the expected output (the target or goals). By establishing output expectations, the completeness and accuracy of the specification is also a part of the targeted improvement. When possible start on a small scale to test possible effects.

DO

Implement the plan, execute the process, make the product. Collect data for charting and analysis in the following "CHECK" and "ACT" steps.

CHECK

Study the actual results (measured and collected in "DO" above) and compare against the expected results (targets or goals from the "PLAN") to ascertain any differences. Look for deviation in implementation from the plan and also look for the appropriateness/completeness of the plan to enable the execution i.e. "Do". Charting data can make this much easier to see trends over several PDCA cycles and in order to convert the collected data into information. Information is what you need for the next step "ACT".

ACT

Request corrective actions on significant differences between actual and planned results. Analyze the differences to determine their root causes. Determine where to

apply changes that will include improvement of the process or product. When a pass through these four steps does not result in the need to improve, the scope to which PDCA is applied may be refined to plan and improve with more detail in the next iteration of the cycle, or attention needs to be placed in a different stage of the process.

The following tools would be used for Coding and Documentation

- Eclipse : Java Coding
- WAMP : Database Server
- Dreamweaver : GUI coding
- Visual Web Ripper : Web Scrapping
- Diafaan : SMS server
- Windows Task Scheduler
- Google Translate
- Rational Unified Process (RUP) : Guidance Suite
- MS Word 2010: Documentation
- MS Project: Project Planning
- Google Docs: Documentation
- Microsoft Windows 7 : Project Development

6. Software Quality Management Plan

The final aim of any project is to come up with quality deliverables that is acceptable to the customer. For this we need to make sure that every intermediate product maintains the quality standard and finally integrates into a quality product. SQAP aims at reaching the desired quality parameters.

6.1 Quality Planning

Defining Quality Standards

Quality of anything is context dependent and it requires a benchmark to prove that the product is quality stuff. However, every quality product has some distinct feature which helps us realize the quality of the product.

- Final product should be error free. One must try to make software as much error free as possible, such that the final product is acceptable to the client.
- The recovery rates from failures must be high.
- The software should do everything specified in the SRS document and it must pass all tests.
- A modular system ensures the error or logical failure of one module does not propagate into other.
- Extra feature always add weight to one's software. It acts as added bonus for client who is always looking for such offers.
- Quality Assurance is followed through the whole Project Life Cycle. Also the variation control ensures good quality software.
- The software must be built in the cost efficient manner.

Measuring Quality Project.

Factors which contributes in measuring Project Quality are-

Review process

Before the document is drafted by the drafting team, a template is adopted in discussion with the Project Manager. 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, follows the template that has been adopted for that particular document. In case any anomalies are identified, the reviewer lists them in a document.

As per IEEE standards is an anomaly is defined as:

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.

All the documents and product being created in the span of the software life cycle is well reviewed by each member of the team. As Plan-Do-Check-Act suggest that after every single document is made, proper reviewing has to be done and then action must be taken to improve the quality of the product. Similarly every single module is being tested and reviewed.

Cost

The software is developed in the planned time without any overheads.

6.2 Quality Assurance Plan and Quality Control

QC is related to the product quality and quality assurance is related to the process quality. If the organization focused on the quality assurance then ultimate product will be of “High quality”. Hence the Team will focus on the quality assurance.

To ensure that the final product is of high quality, some quality control activities must be performed throughout the development and its main purpose is to identify defects. The purpose of a QAP is to specify the activities that need to be performed for identifying and removing defects, and the tools and methods that may be used for that purpose.

QAP should specify all steps needed to come up with a quality product. It should specify:

- All the steps and work needed to complete a project in timely and efficient manner while at the same time not compromising on the quality of the product.
- Activities that must be performed to ensure quality of intermediate product

To ensure that the delivered software is of good quality, it is essential to ensure that all work products like requirements specification, design and test plan are also of good quality. 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. For quality assurance we took the technical reviews and the code reviews at regular intervals and shall make sure that all the deadlines are met.

7. Software Documentation

Every work is properly documented from the very beginning of the project so that necessary information can be obtained at different phases, later during the project development. In case of a sudden disaster, the team can continue working on the project by referring to preserved documents. Changes made during the development of the project can be maintained and monitored easily with the help of proper documentation. If any member of the team needs to temporarily leave the project, he/she can catch up with the work flow and can quickly follow the project by referring to the preserved documents and thus saves a lot of time. It also helps in identifying an error and resolving it. In short, proper documentation controls the development of software.

Documents required:

- Feasibility Report
- Project Plan
- SRS
- Draft User Manual
- SDS
- Test Plan
- Coding Standards
- Test Report
- Review Documents
- Termination Analysis

Document Conventions

S No	Conventions	Metrics	Inference
1	[Heading]	Font size: 16 Font type: Arial Font style: Bold	Heading
2	[Heading]	Font size: 12 Font type: Arial Font style: Bold	Sub-Heading
3	[Heading]	Font size: 12 Font type: Arial	Sub-Sub-Heading

		Font style: Regular	
4	[Heading]	Font size: 11 Font type: Arial Font style: Regular	Body

8. Review Reports

Project Plan Review

The project plan is reviewed in order to ensure that the project is on the right path i.e. according to plan and meeting the deadlines.

Feasibility Report Review

The feasibility report is reviewed in order to confirm that the successful execution as well as completion of the project is feasible and that the planning done earlier in the project is correct and reliable.

Software Requirements Review

The software requirement is reviewed to ensure that the requirements specified by the client in the functional and non-functional requirements have been taken care of.

System design Review

The Technical Design review is held to evaluate the technical adequacy of the software design and the acceptability of the design to satisfy the functional requirements.

Coding Review

The coding review is done at the end of the project to assess the features implemented in the software and recommends appropriate action to be taken. It is done in two ways:

Walkthroughs

This is an informal technique of reviewing and is based on common sense.

Inceptions

This is a formal technique of reviewing and checks for the correctness of individual features developed during the course of the project

Quality Check Points

There would be Quality Checkpoints at the end of various phases, to assure that the deliverables are of acceptable quality. It is planned that the products at the end of each phase in the lifecycle of the software would be reviewed by the team, 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. A Quality Control Manager would be appointed who would ensure that suggestions are incorporated well within the deadline.

9. Validation and Verification

Verification is done to determine whether one phase of software conforms to its previous phase. It concentrates mostly on the phase containment of errors. To ensure quality, we must:

- Evaluate requirements gathered during the requirement phase for the correctness, consistency and accuracy.
- Be sure that there are no anomalies in the requirements gathered. Assess whether the SRS satisfies the project plan.
- Prepare a requirement traceability matrix.

Validation is a process to confirm whether the fully developed software conforms to its requirements. It concentrates mainly on making the final software error free. It also verifies that the final product matches the requirements specified in the SRS document. For validation purpose, a proper error message is displayed and asks for corrective measures in case of an error.

10. Problem Reporting and Corrective Actions

Problem reporting and actions are very important to achieve an error/bug free system. The ultimate target of this is to establish a process that continuously improves the safety and reliability of the system. When a document is reviewed, its quality and correctness is checked and if any changes or modifications are required, they are made accordingly. Even at later stage of the project development, we might discover additions or modifications needed in a document and thus another version of it is released and the updated document is penned in the review document.