Software Quality Management

tutorialspoint

SIMPLY EASY LEARNING

www.tutorialspoint.com

https://www.facebook.com/tutorialspointindia

https://twitter.com/tutorialspoint

About the Tutorial

Software Quality Management is a process that ensures the required level of software quality is achieved when it reaches the users, so that they are satisfied by its performance. The process involves quality assurance, quality planning, and quality control.

This tutorial provides a complete overview of Software Quality Management and describes the various steps involved in the process. The entire content is divided into sections for easy understanding.

Audience

This tutorial is designed for software development professionals so that they can understand the importance of software quality management. It is especially beneficial for software quality managers, software testing professionals, and software developers.

Prerequisites

To get the most out of this tutorial, it is good to have a basic understanding of the Software Development Life Cycle (SDLC).

Copyright & Disclaimer

Table of Contents

1. SQM – Introduction

Quality software refers to a software which is reasonably bug or defect free, is delivered in time and within the specified budget, meets the requirements and/or expectations, and is maintainable. In the software engineering context, software quality reflects both **functional quality** as well as **structural quality**.

- **Software Functional Quality** – It reflects how well it satisfies a given design, based on the functional requirements or specifications.

- **Software Structural Quality** – It deals with the handling of non-functional requirements that support the delivery of the functional requirements, such as robustness or maintainability, and the degree to which the software was produced correctly.

- **Software Quality Assurance** – Software Quality Assurance (SQA) is a set of activities to ensure the quality in software engineering processes that ultimately result in quality software products. The activities establish and evaluate the processes that produce products. It involves process-focused action.

- **Software Quality Control** – Software Quality Control (SQC) is a set of activities to ensure the quality in software products. These activities focus on determining the defects in the actual products produced. It involves product-focused acction.

The Software Quality Challenge

In the software industry, the developers will never declare that the software is free of defects, unlike other industrial product manufacturers usually do. This difference is due to the following reasons.

Product Complexity

It is the number of operational modes the product permits. Normally, an industrial product allows only less than a few thousand modes of operation with different combinations of its machine settings. However, software packages allow millions of operational possibilities. Hence, assuring of all these operational possibilities correctly is a major challenge to the software industry.

Product Visibility

Since the industrial products are visible, most of its defects can be detected during the manufacturing process. Also the absence of a part in an industrial product can be easily detected in the product. However, the defects in software products which are stored on diskettes or CDs are invisible.

Product Development and Production Process

In an industrial product, defects can be detected during the following phases:

- **Product development:** In this phase, the designers and Quality Assurance (QA) staff checks and tests the product prototype to detect its defects.

- **Product production planning**: During this phase, the production process and tools are designed and prepared. This phase also provides opportunities to inspect the product to detect the defects that went unnoticed during the development phase.

- **Manufacturing**: In this phase, QA procedures are applied to detect failures of products themselves. Defects in the product detected in the first period of manufacturing can usually be corrected by a change in the product's design or materials or in the production tools, in a way that eliminates such defects in products manufactured in future.

However, in the case of software, the only phase where defects can be detected is the development phase. In case of software, product production planning and manufacturing phases are not required as the manufacturing of software copies and the printing of software manuals are conducted automatically.

The factors affecting the detection of defects in software products versus other industrial products are shown in the following table.

Characteristic	Software products	Other industrial products
Complexity	Millions of operational options	thousand operational options
Visibility of product	Invisible Product Difficult to detect defects by sight	Visible Product Effective detection of defects by sight
Nature of development and production process	Can detect defects in only one phase	Can detect defects in all of the following phases ✓Product development ✓Product production planning ✓Manufacturing

These characteristics of software such as complexity and invisibility make the development of software quality assurance methodology and its successful implementation a highly professional challenge.

2. SQM – Software Quality Factors

The various factors, which influence the software, are termed as software factors. They can be broadly divided into two categories. The first category of the factors is of those that can be measured directly such as the number of logical errors, and the second category clubs those factors which can be measured only indirectly. For example, maintainability but each of the factors is to be measured to check for the content and the quality control.

Several models of software quality factors and their categorization have been suggested over the years. The classic model of software quality factors, suggested by McCall, consists of 11 factors (McCall *et al*., 1977). Similarly, models consisting of 12 to 15 factors, were suggested by Deutsch and Willis (1988) and by Evans and Marciniak (1987).

All these models do not differ substantially from McCall's model. The McCall factor model provides a practical, up-to-date method for classifying software requirements (Pressman, 2000).

McCall's Factor Model

This model classifies all software requirements into 11 software quality factors. The 11 factors are grouped into three categories – product operation, product revision, and product transition factors.

- **Product operation factors:** Correctness, Reliability, Efficiency, Integrity, Usability.
- **Product revision factors:** Maintainability, Flexibility, Testability.
- **Product transition factors:** Portability, Reusability, Interoperability.

Product Operation Software Quality Factors

According to McCall's model, product operation category includes five software quality factors, which deal with the requirements that directly affect the daily operation of the software. They are as follows:

Correctness

These requirements deal with the correctness of the output of the software system. They include:

- Output mission
- The required accuracy of output that can be negatively affected by inaccurate data or inaccurate calculations.
- The completeness of the output information, which can be affected by incomplete data.
- The up-to-dateness of the information defined as the time between the event and the response by the software system.
- The availability of the information.

- The standards for coding and documenting the software system.

Reliability

Reliability requirements deal with service failure. They determine the maximum allowed failure rate of the software system, and can refer to the entire system or to one or more of its separate functions.

Efficiency

It deals with the hardware resources needed to perform the different functions of the software system. It includes processing capabilities (given in MHz), its storage capacity (given in MB or GB) and the data communication capability (given in MBPS or GBPS).

It also deals with the time between recharging of the system's portable units, such as, information system units located in portable computers, or meteorological units placed outdoors.

Integrity

This factor deals with the software system security, that is, to prevent access to unauthorized persons, also to distinguish between the group of people to be given read as well as write permit.

Usability

Usability requirements deal with the staff resources needed to train a new employee and to operate the software system.

Product Revision Quality Factors

According to McCall's model, three software quality factors are included in the product revision category. These factors are as follows:

Maintainability

This factor considers the efforts that will be needed by users and maintenance personnel to identify the reasons for software failures, to correct the failures, and to verify the success of the corrections.

Flexibility

This factor deals with the capabilities and efforts required to support adaptive maintenance activities of the software. These include adapting the current software to additional circumstances and customers without changing the software. This factor's requirements also support perfective maintenance activities, such as changes and additions to the software in order to improve its service and to adapt it to changes in the firm's technical or commercial environment.

Testability

Testability requirements deal with the testing of the software system as well as with its operation. It includes predefined intermediate results, log files, and also the automatic diagnostics performed by the software system prior to starting the system, to find out whether all components of the system are in working order and to obtain a report about the detected faults. Another type of these requirements deals with automatic diagnostic checks applied by the maintenance technicians to detect the causes of software failures.

Product Transition Software Quality Factor

According to McCall's model, three software quality factors are included in the product transition category that deals with the adaptation of software to other environments and its interaction with other software systems. These factors are as follows:

Portability

Portability requirements tend to the adaptation of a software system to other environments consisting of different hardware, different operating systems, and so forth. The software should be possible to continue using the same basic software in diverse situations.

Reusability

This factor deals with the use of software modules originally designed for one project in a new software project currently being developed. They may also enable future projects to make use of a given module or a group of modules of the currently developed software. The reuse of software is expected to save development resources, shorten the development period, and provide higher quality modules.

Interoperability

Interoperability requirements focus on creating interfaces with other software systems or with other equipment firmware. For example, the firmware of the production machinery and testing equipment interfaces with the production control software.

3. SQM – SQA Components

Software Quality Assurance (SQA) is a set of activities for ensuring quality in software engineering processes. It ensures that developed software meets and complies with the defined or standardized quality specifications. SQA is an ongoing process within the Software Development Life Cycle (SDLC) that routinely checks the developed software to ensure it meets the desired quality measures.

SQA practices are implemented in most types of software development, regardless of the underlying software development model being used. SQA incorporates and implements software testing methodologies to test the software. Rather than checking for quality after completion, SQA processes test for quality in each phase of development, until the software is complete. With SQA, the software development process moves into the next phase only once the current/previous phase complies with the required quality standards. SQA generally works on one or more industry standards that help in building software quality guidelines and implementation strategies.

It includes the following activities:

- Process definition and implementation
- Auditing
- Training

Processes could be:

- Software Development Methodology
- Project Management
- Configuration Management
- Requirements Development/Management
- Estimation
- Software Design
- Testing, etc.

Once the processes have been defined and implemented, Quality Assurance has the following responsibilities:

- Identify the weaknesses in the processes
- Correct those weaknesses to continually improve the process

Components of SQA System

An SQA system always combines a wide range of SQA components. These components can be classified into the following six classes:

Pre-project components

This assures that the project commitments have been clearly defined considering the resources required, the schedule and budget; and the development and quality plans have been correctly determined.

Components of project life cycle activities assessment

The project life cycle is composed of two stages: the development life cycle stage and the operation–maintenance stage.

The development life cycle stage components detect design and programming errors. Its components are divided into the following sub-classes: Reviews, Expert opinions, and Software testing.

The SQA components used during the operation–maintenance phase include specialized maintenance components as well as development life cycle components, which are applied mainly for functionality to improve the maintenance tasks.

Components of infrastructure error prevention and improvement

The main objective of these components, which is applied throughout the entire organization, is to eliminate or at least reduce the rate of errors, based on the organization's accumulated SQA experience.

Components of software quality management

This class of components deal with several goals, such as the control of development and maintenance activities, and the introduction of early managerial support actions that mainly prevent or minimize schedule and budget failures and their outcomes.

Components of standardization, certification, & SQA system assessment

These components implement international professional and managerial standards within the organization. The main objectives of this class are utilization of international professional knowledge, improvement of coordination of the organizational quality systems with other organizations, and assessment of the achievements of quality systems according to a common scale. The various standards may be classified into two main groups: quality management standards and project process standards.

Organizing for SQA – the human components

The SQA organizational base includes managers, testing personnel, the SQA unit and the persons interested in software quality such as SQA trustees, SQA committee members, and SQA forum members. Their main objectives are to initiate and support the implementation of SQA components, detect deviations from SQA procedures and methodology, and suggest improvements.