Software Quality Management

CONTENTS

I. Basic Quality Concepts
II. Software Quality Assurance (SQA)
   1. Definition of SQA
   2. SQA Activities
III. Quality Evaluation Standards
   1. Six sigma for software
   2. ISO 9000 for software - concept and major considerations
IV. CMMI - CMMI Levels, Process Areas considered.
V. CMMI Vs ISO.
VI. McCall's Quality factors.
I. Basic Quality Concepts

1. Quality means that a product satisfies the demands of its specifications
2. It also means achieving a high level of customer satisfaction with the product
3. In software systems this is difficult
   i. Customer quality requirements (e.g. efficiency or reliability) often conflict with developer quality requirements (e.g. maintainability or reusability)
   ii. Software specifications are often incomplete, inconsistent, or ambiguous

II. Software Quality Assurance (SQA)

1. Definition of SQA
   (Question: Explain SQA with its activities. – 4 Marks)
   i. Conformance to software requirements is the foundation from which software quality is measured.
   ii. Specified standards are used to define the development criteria that are used to guide the manner in which software is engineered.
   iii. Software must conform to implicit requirements (ease of use, maintainability, reliability, etc.) as well as its explicit requirements.

2. SQA Activities

   These activities are performed (or facilitated) by an independent SQA group that:
   i. Prepares an SQA plan for a project.
   ii. Participates in the development of the project’s software process description.
   iii. Reviews software engineering activities to verify compliance with the defined software process.
   iv. Audits designated software work products to verify compliance with those defined as part of the software process.
   v. Ensures that deviations in software work and work products are documented and handled according to a documented procedure.
   vi. Records any noncompliance and reports to senior management.

III. Quality Evaluation Standards

1. Six sigma for software - Concept of DMAIC and DMDAV Approach
   (Question: Explain six sigma for project development. - 6 Marks)
   i. Six Sigma through the correct application of statistical tools can reap a company enormous rewards that will have a positive effect for years
ii. Six Sigma can be a dismal failure if not used correctly.
iii. A disciplined quantitative approach for improvement of defined metrics
iv. Can be applied to all business processes, manufacturing, finance and services

Figure 1: Six Sigma Roadmap

v. The failure rate of any project is represented as the six sigma with failure rate as shown in Figure 2.

Figure 2: Sig sigma Failure rate
2. ISO 9000 for software - concept and major considerations

(Question: Explain the ISO 9000 standards for software development. – 4 Marks)

i. International set of standards for quality management

ii. Quality standards and procedures must be documented in an organizational quality manual

iii. An external body is often used to certify that the quality manual conforms to ISO 9000 standards

iv. Many customers are demanding that suppliers are ISO 9000 certified

---

IV. CMMI - CMMI Levels, Process Areas considered.

(Question: Explain CMMI models with diagram. – 6 Marks)

1. CMMI (Capability Maturity Model Integration) is a proven industry framework to improve product quality and development efficiency for both hardware and software

   i. Sponsored by US Department of Defence in cooperation with Carnegie Mellon University and the Software Engineering Institute (SEI)

   ii. Many companies have been involved in CMMI definition such as Motorola and Ericsson
iii. CMMI has been established as a model to improve business results

2. CMMI, staged, uses 5 levels to describe the maturity of the organization, same as predecessor CMM
   i. Vastly improved version of the CMM
   ii. Emphasis on business needs, integration and institutionalization

V. CMMI Vs ISO.

- The comparison of CMMI vs ISO reveals that while CMMI is more focused, complex, and aligned with business objectives, ISO is flexible, wider in scope and not directly linked to business objectives.
- The attainment of either a CMMI ranking or ISO certification nevertheless help organizations establish a quality management system and focus on continuous improvement.

1. CMMI vs ISO: Conceptual Difference

Anuradha Bhatia
i. The fundamental difference between CMMI vs ISO is conceptual. CMMI is a process model and ISO is an audit standard.

ii. CMMI is a set of related "best practices" derived from industry leaders and relates to product engineering and software development.

iii. Businesses receive CMMI ratings from Level 1 to Level 5 depending upon the extent of compliance to key performance areas specified in the selected CMMI process area.

iv. ISO is a certification tool that certifies businesses whose processes conform to the laid down standards.

2. CMMI vs ISO: Scope
   i. CMMI is rigid and extends only to businesses developing software intensive systems. ISO is flexible and applicable to all manufacturing industries.
   
   ii. CMMI focuses on engineering and project management processes whereas ISO’s focus is generic in nature.
   
   iii. CMMI mandates generic and specific practices and businesses have a choice of selecting the model relevant to their business needs from 22 developed process areas.
   
   iv. ISO requirements are same for all companies, industries, and disciplines.

3. CMMI vs ISO: Approach
   i. CMMI requires ingraining processes into business needs so that such processes become part of corporate culture and do not break down under the pressure of deadlines.
   
   ii. ISO specifies to conformance and remains oblivious as to whether such conformance is of strategic business value or not.
   
   iii. CMMI approaches risk management as an organized and technical discipline by identifying risk factors, quantifying such risk factors, and tracking them throughout the project life cycle.
   
   iv. ISO was until recently neutral on risk management. ISO 31000:2009 now provides generic guidelines for the design, implementation, and maintenance of risk management processes throughout an organization.
   
   v. Although CMMI focuses on linkage of processes to business goals, customer satisfaction is not a factor in the ranking whereas customer satisfaction is an important part of ISO requirements.

4. CMMI vs ISO Implementation
   i. Neither CMMI nor ISO requires the establishment of new processes.
ii. CMMI compares the existing processes to industry best practices whereas ISO requires adjustment of existing processes to confirm to the specific ISO requirements.

iii. In practice, some organizations tend to rely on extensive documentation while implementing both CMMI and ISO.

iv. Most organizations tend to constitute in-house teams, or rely on external auditors to see through the implementation process.

VI. **McCall's Quality factors**

*(Question: Explain McCall's Quality factor with diagram. - 8 Marks)*

The factors that affect S/W quality can be categorized in two broad groups:

1. factors that can be directly measured (defects uncovered during testing)
2. factors that can be measured only indirectly (Usability and maintainability)

![McCall's Quality Factor Diagram](image)

The S/W quality factors shown above focus on three important aspects of a S/W product:

i. Its operational characteristics
ii. Its ability to undergo change
iii. Its adaptability to new environments

**The various factors of quality are:**

i. **Correctness**: The extent to which a program satisfies its specs and fulfills the customer’s mission objectives.
ii. **Reliability**: The extent to which a program can be expected to perform its intended function with required precision.

iii. **Efficiency**: The amount of computing resources and code required to perform a function.

iv. **Integrity**: The extent to which access to S/W or data by unauthorized persons can be controlled.

v. **Usability**: The effort required to learn, operate, prepare input for, and interpret output of a program.

vi. **Maintainability**: The effort required to locate and fix errors in a program.

vii. **Flexibility**: The effort required to modify an operational program.

viii. **Testability**: The effort required to test a program to ensure that it performs its intended function.

ix. **Portability**: The effort required to transfer the program from one hardware and/or software system environment to another.

x. **Reusability**: The extent to which a program can be reused in other applications-related to the packaging and scope of the functions that the program performs.

xi. **Interoperability**: The effort required to couple one system to another.