

Software Project Quality Management

Kunal Jamsutkar¹, Viki Patil², P. M. Chawan³

(Department of Computer Science, VJTI, MUMBAI, INDIA)

ABSTRACT

Quality Management is very important in Software Projects. Quality is achieved to the extent that a project end product meets the client's needs and expectations; it also means that a product should meet its specification. Paper discusses about Life Cycle approach to Software Quality Management process and its principles, activities, factors, methods, benefits, and also Principles of Quality Risk Management.

Keywords — **Clients needs and expectations, project management, quality control, quality factors.**

1. Introduction

ISO 8402 defines Quality as “the totality of features and characteristics of a product or a service that bear on its ability to satisfy the stated and implied needs” [1].

The quality principles that have proven so successful in software project business are fully applicable to project management [1]. A Quality Management System can be defined as: “A set of coordinated activities to direct and control an organization in order to continually improve the effectiveness and efficiency of its performance”. As per international standards Quality comprises all characteristics and significant features of a product or an activity which relate to the satisfying of given requirements. Quality is not poured into an end product on the very last day of the project. Quality results from the progressive and cumulative contribution of the many project activities performed throughout the project's life cycle [1]. Software quality is the extent to which an industry-defined set of desirable features are incorporated into a product so as to enhance its lifetime performance.

2. Software Quality Management process

A Quality Management Process is a set of procedures that are followed to ensure that the deliverables produced by a team are "fit for purpose". The start of the Quality Management Process involves setting quality targets, which are agreed with the customer.

Quality Management Process will help us to:

- Set Quality Targets to be met by your team.
- Define how those quality targets will be measured.
- Take the actions needed to measure quality.
- Identify quality issues and improvements.

- Report on the overall level of quality achieved.

3. Software Quality Management Activity

In Project Quality management there are three processes involved as follows:

3.1. Quality planning

It means first select the applicable procedures and standards for a particular project and then modify these as required. It is required to explain important factor that contribute the customer requirements.

3.2. Quality assurance

It establishes organizational procedures and standards for quality. It involves evaluating overall project performance to ensure that the project will fulfill the relevant quality standards. It is taking responsibility for quality through project's life cycle.

3.3 Quality control

It ensured that procedures and standards are implemented by the software development team. It is monitoring specific project results to ensure that improve while identifying ways the overall quality according to quality standards.

4. Software Quality Management Models

Quality models acts as a bridge between metrics and characteristics in the software. According to Wallmüller (1994) "one of the oldest and most frequently used software Quality models are that of McCall *et al.* (1979). There are other models such as that of Murine & Carpenter (1984) are derived from it.

McCall's model is used in the United States for very large projects in the military, space and public domain. It was developed in 1976-7 by the US Airforce Electronic System Division (ESD), the Rome Air Development Centre (RADDC) and General Electric (GE) with the aim of improving the quality of software products. One explicit aim was to make quality measurable.

McCall had start with a volume of 55 quality characteristics called as factors which have an important influence on quality. For simplicity, McCall then reduced the number of characteristics to the following eleven: It is categorized into two parts as External Quality and Internal Quality.

4.1 External Quality

Features are: Integrity, Reliability, Usability, and Accuracy.

4.2 Internal Quality

Features are: Efficiency, Testability, Maintainability, Flexibility, Portability, Re-usability, and Transferability.

5. Software Quality Factors

Eleven Factors affecting software quality are:

5.1 Efficiency

McCall's view of efficiency or performance is concerned with the efficient use of computer code to perform processes and the efficient use of storage resources. There are a number of techniques that can be used to achieve both of these objectives. Types are:

5.1.1 Programming languages

Selecting the most appropriate programming language for the problem has a major impact on program efficiency. For example, business applications which require substantial volumes of reports might best be programmed in COBOL while programs requiring substantial scientific calculations might be best accommodated by FORTRAN.

5.1.2 Operating systems

Modern operating systems have the ability to perform multi-tasking thereby improving system performance by facilitating background operations.

5.1.3 Design

Strategies that address cohesion and coupling, normalisation techniques to reduce data redundancy and algorithms that optimise process time should always be employed.

5.1.4 Access strategies

Algorithms that optimise seek time, rotational delay and data transfer time must be continuously searched out and implemented to improve the following:

- Keeping local variables within procedures.
- Good use of parameter passing.
- Meaningful variable and procedure names.
- Proper documentation efficiency.

5.1.5 Programming techniques

Typical good programming techniques and practices like

- Top-down design for complex problems
- Sequence, selection and iteration constructs

5.2 Testability

It means that the ability to verify requirements. Testability is a measure of how well

system or components allow you to create test criteria and execute tests to determine if the criteria are met.

5.3 Maintainability

It refers to easily and inexpensively the maintenance tasks can be performed. Maintainability is the ability of the system to undergo changes with a degree of ease. These changes could impact components, services, features, and interfaces when adding or changing the application's functionality in order to fix errors, or to meet new business requirements.

5.4 Flexibility

It is reflected in the cost of modifying an operational system.

5.5 Portability

It means that a software system can be easily adopted to run in a different execution environment.

5.6 Reusability

It is the probability that a component will be used in other components or scenarios to add new functionality with little or no change. Reusability minimizes the duplication of components and the implementation time.

5.7 Interoperability

It is the ability of a system or different systems to operate successfully by communicating and exchanging information with other external systems written and run by external parties.

5.8 Integrity

System integrity refers to extent to which access to software or secure data by unauthorised persons. It is play important role in network based application.

5.9 Reliability

It is difficult to construct large software system which is correct. It might possible that few functions may not work in all scenarios, therefore software consider as incorrect. However the software still accepts able the failure rate is very small and it does affect so much.

5.10 Usability

It means software system is to be easy to use for human user. Normally client much emphasis on the user interfaces of software system.

5.11 Correctness

Software system is expected to satisfy its specification and fulfils the user's mission objectives.

6. The Quality of the Project End Product

6.1 As per ISO's definition of quality to a project end product implies that:

- The client's point of view should prevail always when assessing quality.
- Quality depends on a host of features and characteristics that contribute, to various degrees, to the client's needs and expectations.
- Quality is accrued progressively throughout the project life cycle.
- Corrective actions should be initiated as soon as significant quality deviations are detected.

6.2 Method for assessing and controlling quality

Therefore, a method for assessing and controlling quality must meet the following four requirements:

6.2.1 The assessment method must enable project managers to elucidate and structure the needs and expectations of the client. Quality is considered by many as a measure of the client's satisfaction. Overall client satisfaction can be decomposed into a hierarchical structure of quality criteria. This is performed through a top-down process whereby the more general objectives are decomposed into lower-level objectives of greater detail.

6.2.2 The method must be capable of assessing and aggregating lower level quality criteria into higher level quality objectives. This might prove to be quite difficult in the case of qualitative criteria necessitating subjective evaluations. The project manager and the client must therefore agree on a formal way to quantify their subjective values.

6.2.3 The method must provide a means of assessing the planned and earned quality of the project end product throughout its life cycle. Quality features and characteristics are achieved in a progressive and cumulative fashion as the project is being executed. The method must therefore link the project activities to the quality attributes of the project end product.

6.2.4 Finally, the quality assessment and control method should provide measures of quality deviations to enable project managers to initiate corrective actions early. Indeed, the longer it takes to detect and correct quality deviations, the more it costs to have the work redone. Because deviations can be due to randomness in the data, a threshold between the expected quality level and the estimated quality level must be established. Exceeding the threshold should initiate corrective actions. Consequently, quality deviations should be traced to specific project activities.

7. Software Quality Management Lifecycle Approach

Application quality management should focus on four key processes to optimize lifecycle quality.

LQM encompasses the following four quality processes: define application quality goals and the metrics against which quality will be measured; measure application quality; manage quality holistically throughout the application lifecycle; and improve quality on an ongoing basis.

Define application quality goals and metrics using an agreed-upon set of criteria and standard methodology for defining what success means in your business:

- Define desired levels of quality.
- Measure current quality to set a baseline.
- Determine quality metrics based on business goals and drivers.

Measure application quality status and quality progress, versus the metrics set as quality benchmarks:

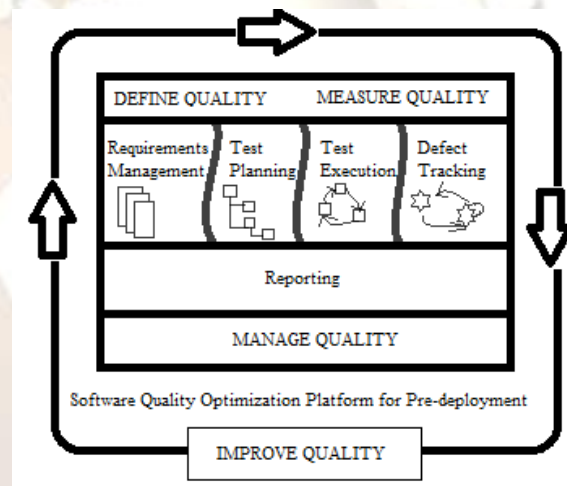


Fig. 1. Software quality life cycle

Capture data on previously defined quality metrics. It Measures and track quality metrics important to varied constituents. Determine status of application quality against quality goals. Prove compliance with regulatory requirements. Prove an application's release readiness to support business goals. Manage all aspects of software quality holistically to ensure continuous quality at the "speed of change". Integrate people, processes and technology through an architected common quality platform. Manage the entire quality process from end-to-end. Obtain up-to-the-minute data to enable sound decision making about resource allocation, schedule and overall release readiness. Gain visibility of software projects throughout all phases of the software application lifecycle.

Dynamically adapt to meet changing business situations, emerging technologies and/or new regulatory challenges. Improve development efforts as you gain knowledge. The use of improved quality practices needs to be reintroduced into the process for potential benefits to be realized.

Continuously improve the release process, extending quality to encompass previously uncovered areas. Meet time-to-market goals by introducing new efficiencies into the quality process. Reuse existing test assets, leveraging resources and producing ever-higher quality software. Grow the commitment to quality systemically, throughout the corporate culture.

An organization that embraces LQM blends business and software development activities into an integrated business practice, maximizing its ability to attain stated business objectives. The supporting centralized infrastructure eliminates multiple points of exposure and fosters repeatable, reusable quality assets.

8. Cost to Quality

The Five Cost Factors Related to Quality is as follows:

8.1 Prevention cost

It is cost of planning and executing a project so it is error-free or within an acceptable error range.

8.2 Appraisal cost

It is cost of evaluating processes and their outputs to ensure quality.

8.3 Internal failure cost

It is cost incurred to correct an identified defect before the customer receives the product.

8.4 External failure cost

It is cost that relates to all errors not detected and corrected before delivery to the customer Measurement and test.

8.5 Equipment costs

It is capital cost of equipment used to perform prevention and appraisal activities.

9. Principles of Quality Risk Management

Primary principles of quality risk management are:

9.1 Evaluation of the risk to quality should be based on scientific knowledge and ultimately link to the protection of the patient

9.2 The level of effort, formality and documentation of the quality risk management process should be commensurate with the level of risk.

General quality risk management process Quality risk management is a systematic process for the assessment, control, communication and review of risks to the quality of the product across the product lifecycle.

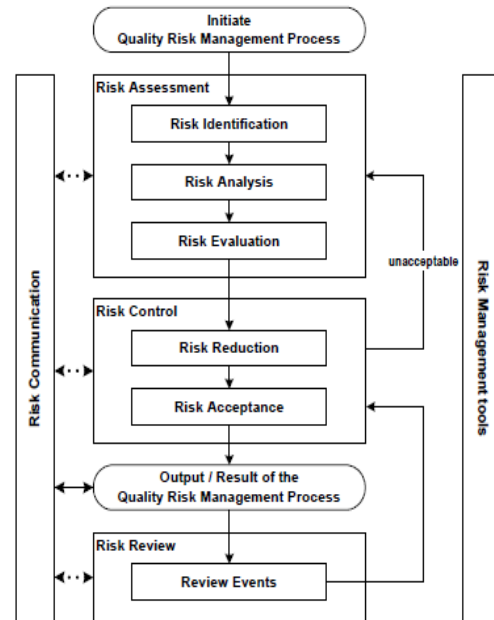


Fig. 2. Model for quality risk management

A model for quality risk management is outlined in the Figure 2. Other models could be used. The emphasis on each component of the framework might differ from case to case but a robust process will incorporate consideration of all the elements at a level of detail that is commensurate with the specific risk.

10. Benefits of Quality Management

A fully documented QMS will ensure that two important requirements are met:

10.1 Customers' requirements

Needs Confidence in the ability of the organisation to deliver the desired product and service consistently meeting their needs and expectations.

10.2 Organisation's requirements

Both internally and externally, and at an optimum cost with efficient use of the available resources materials, human, technology and information.

11. Conclusion

Lifecycle quality management initiatives help companies move their software quality efforts into the 21st century and to more effectively align application quality to support business needs such as faster time-to-market, improved customer satisfaction and competitive advantage. Software Project Quality

Management will ensure to meet the customer's and organisation's requirements. A "Quality Assurance Process" and "Quality Control Process" are then undertaken after quality planning, to measure and report the actual quality of deliverables. As the part of Quality Management Process, any quality issues are identified and resolved quickly.

REFERENCES

Journal Papers

- [1] Jean Paul Paquin, Jean Couillard, and Dominique J. Ferrand, Assessing and Controlling the Quality of a Project End Product: The Earned Quality Method, *IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT, VOL. 47, NO. 1, FEBRUARY 2000* White Papers
- [2] Borland the open ALM Company, Software Quality Management, *white paper April 2006* Books
- [3] Staffordshire University, (*Software Quality: Definitions and Strategic Issues*), School of Computing Report, Ronan Fitzpatrick, April 1996. Websites
- [4] www.dti.gov.uk/quality/qms