

# Software Quality Management for Improving Quality of Software Product

Shubham Srivastava<sup>1</sup>, Prabha singh<sup>2</sup>, Chayanika Srivastava<sup>3</sup>

<sup>1,2,3</sup>Department of Computer Science and Engineering, ITM, GIDA, Gorakhpur (India)

**Abstract** – Software quality management (SQM) is a process that ensures that developed software meets and complies with defined or standardized quality specifications. SQM is an ongoing process within the software development life cycle (SDLC) that routinely checks the developed software to ensure it meets desired quality measures.

In this article, it is highlighted that how to combine software quality management life cycle with software development life cycle to generate better quality and outcomes. This article takes a different approach by examining how software quality management practices can impact project outcomes. Software quality management (SQM) plays a critical role in the software development lifecycle (SDLC) and can impact a project's overall success.

**Index Terms** – SQM, SQA, SQP, SQC and SDLC.

## 1. INTRODUCTION

Today's scenario, it is really difficult to implement successful IT projects which is a critical strategic method for entire industrial sectors. This is even more important in times of scarce resources needed for other competing strategic initiatives important to a firm. A lot of software projects still fail to deliver on time and within target costs and other necessary specifications. Software quality management (SQM) is a management process the goal of which is to develop and manage the quality of a software to make sure the product satisfies the user. SQM is a quality management culture in an organization where quality is viewed and maintained by everyone. Software Quality Management (SQM) includes different layer such as Software Quality Assurance (SQA) layer, Software Quality Plan (SQP) layer and Software Quality Control (SQC) layer. The Software quality assurance (SQA) is a monitoring process which is used to ensure the quality in entire software development lifecycle process. It is also a continuous assessment mechanism which facilitates certain procedures for project development with specific standards along with documentation. The procedures should be used to assure quality outcome (zero defects) and project success. The Software Quality Plan (SQP) is an initial project level quality plan for declaring project commitment and SQP contain quality goals to be achieved, expected risks and risk management.

The Software Quality Control (SQC) ensures in-process that both SQA and SQP are being followed by the development teams.



Fig:-1 SQM ACTIVITIES

## 2. PROJECT MANAGEMENT FAILURES

Some Basic Reasons for Failure:-

- Planning: Insufficient planning
- Scope: Poorly defined and requirement and scope
- Stakeholder: Ineffective stakeholder management, Lack of user and stakeholder involvement
- Commitment: Lack of executive support or project sponsorship
- Time: Poor estimation and scheduling, unrealistic deadlines
- Product: Speed to market, desire for innovation, correct errors
- Project goal: Unrealistic project goal
- Change control: Poorly managed change control
- Technology: Change in technology
- Cost: Unrealistic budget
- Quality and testing: Elimination of quality assurance practice, inadequate testing or reduction of testing

time within the software development lifecycle

- Resources: Insufficient qualified resource
- Communication: Poor and ineffective communication
- Risk: Insufficient risk planning and risk management
- Politics: Stakeholder politics, user politics, corporate politics
- Project manager: Poor project management or project manager
- Process: Undefined development process or lack of adherence to process standard, failure to implement best practices and lesson learned

There are some challenges related to project development such as technology used for project development, the process used for project development, scope and complexity of developed projects, and risk of developed projects. Project success requires carefully followed processes that assure quality at every phase of the project and development lifecycles. Quality assurance practices can minimize project failures by providing checks throughout this process.

### 3. SOFTWARE QUALITY ASSURANCE

A software quality assurance is a mechanism to assure the acceptance of quality of software along with an organization's procedures and standards. SQA is an ongoing process within the software development life cycle (SDLC) that routinely checks the developed software to ensure it meets desired quality measures.



Fig:-2 Quality Assurance component

It is process which is integrated with project management as well as the project development lifecycles to check control mechanism along with software standards and procedures. The objective of SQA is to reduce risk and improve quality of the project of software with time and cost constraints. SQA generally works on one or more industry standards that help in building software quality guidelines and implementation strategies.

### 4. SQA PROCESS AND ROLE OF SQA TEAMS

The SQA is a monitoring process which is used to ensure the quality in entire software development lifecycle process. It is also a continuous assessment mechanism which facilitates certain procedures for project development with specific standards along with documentation. The procedures should be used to assure quality outcome (zero defects) and project success. At a high level, the function of SQA is to perform the following:

- Software project planning: Quality practices should be planned in advance which can be implemented further.
- User requirement: Requirements should be checked in entire project development process to satisfy user needs.
- Design process & Coding: Certain methodologies are followed in design process. Coding standard and guidelines must be established and implemented.
- Software integration and Testing: Software integration and testing should be planned and compiled as per requirement.
- Conduct random and scheduled audits: Perform SQA audits to assure the necessary controls are in place.

The SQA process consists of a variety of phases with specific activities. These activities should be performed by a SQA team which is responsible for software quality assurance planning, analysis, and reporting. SQA is most effective when it reports up through a separate management team so they can remain committed to the process and remain objective to the deliverable. The responsibilities of the SQA team include review of documentation for completeness and adherence to standards, participation in inspections, review of test results, and periodic audits of controls.

### 5. SQA PROCESS

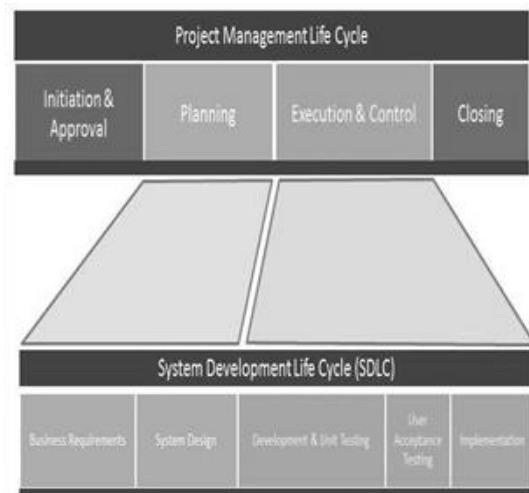


Fig:-3 PQMLC & SDLC

SQA processes provide assurance throughout the project or software management and development lifecycle. The combined project quality management life cycle (PQMLC) and Software development life cycle (SDLC) consists of following phases – initiation, planning, analysis, design, implementation, testing, and closing as shown in fig.3.

#### 6. IMPLICATIONS FOR PRACTICE

SQA process is a complicated process which needs commitment and support from the entire SQA team. SQA process must include metrics to enhance the future improved development of software products or projects. All stakeholders need to understand and agree upon the role of the SQA process and team.

#### 7. CONCLUSION

In this topic, we have discussed the reasons why projects fail and made a case for why a software quality management process is necessary to mitigate risk of failure and reduce failure rates. We have proposed a consolidated definition of SQM and developed a process for assuring software quality that encompasses the whole software development and project management life cycles using

SQA.

Certain basic items must be checked such as

- Purpose of plan and its scope

- Organization structure and SQA task
- Project documents, user and technical document
- Standards, practices and conventions
- Review and audit
- Test plan and procedure
- Error reporting and correcting methods
- Tools, code and supplier control
- Record collection and maintenance
- Risk management

#### REFERENCES

- [1] Agarwal, R. Nayak, P., Manickam, M., Suresh, P., & Modi, N. (2007). Virtual quality assurance facilitation model. IEEE International Conference on Global Software Engineering (ICGSE 2007), IEEE Computer Society.
- [2] Christensen, M. J., & Thayer, R. H. (2001). The project manager's guide to software engineering best practices. Los Alamitos, CA: IEEE Computer Society.
- [3] Galin, D. (2004). Software quality assurance: From theory to implementation. Harlow, UK: Pearson Education Limited.
- [4] Pressman, R. S. (6th Ed.) (2005). Software engineering: A practitioner's approach, Boston: McGraw Hill.s
- [5] Murugesan, S. (1994, Dec. 21-22). Attitude towards testing: A key contributor to software quality. IEEE's Proceedings of 1st International Conference on Software Testing, Reliability and Quality Assurance.