Significance of S/W Quality Assurance in SDLC : A Review

[1.]**Anjana Singh,** [1].Department of Computer Science, A.I.J.H.M.College,
Rohtak (India), ***E-Mail :anjanatalan81@gmail.com***

Abstract :
Quality is the main focus of any software engineering project. Without measuring, we cannot be sure of the level of quality in software. So the methods of measuring the quality are software testing techniques. This thesis report relates various types of testing technique that we can apply in measuring various quality attributes. Software development and maintenance is used to make the error-free Software and also concentrate on time-consuming and complex activity. To evaluate the quality of a software product and to keep its level high is much more difficult than to do them for the other industrial products. For maintaining the quality, performance, speed, efficiency and cost of the software the Software quality Assurance activities, principles and its methods are implemented in the early stages of software engineering development phases. The purpose of testing can be quality assurance, verification, and validation or reliability estimation. It is a tradeoff between budget, time and quality. Software Quality is the central concern of software engineering. Testing is the single most widely used approach to ensuring software quality.

ISSN 2454-308X
9 770024 543081

Index Terms **— Software Quality Assurance, Software Engineering & SDLC etc**

1. INTRODUCTION :

SDLC refers to software development life cycle, i.e. the various stages used in the life cycle of software development. Software Testing is one of the most important phase of Software Development Life Cycle and help to ensure the delivery of a quality software. Software testing is the process of executing a program or system with the intent of finding errors. Software is not unlike other physical processes where inputs are received and outputs are produced. Where software differs is in the manner in which it fails. Quality cannot be achieved by assessing an already completed product. The aim therefore, is to prevent quality defects or deficiencies in the first place, and to make the products assessable by quality assurance measures. Some quality assurance measures include: structuring the development process with a software development standard and supporting the development process with methods, techniques, and tools. The undetected bugs in the software that caused millions of losses to business have necessitated the growth of independent testing, which is performed by a company other than the developers of the system. Quality is frequently associated with cost, meaning that high quality equals high cost. This is confusion between quality of design and quality of conformance.

2. S/W QUALITY ASSURANCE (SQA) :

Software quality assurance (SQA) is defined as a planned and systematic approach to the evaluation of the quality, software product standards, processes and procedures. SQA includes the process of assuring that standards and procedures are established and are followed throughout the software life cycle.

The modern view of quality assurance takes a more sophisticated view than that of fitness for purpose. A high quality product is one which has associated with it a number of quality factors

The quality factor is categorized into three types :

- Portability
- Usability
- Reusability

3. TECHNICAL BEHAVIOR OF SQA :

The technical activities of Software Quality Assurance system are following:

- **Requirement Analysis:** This is the process of discovering the requirements that the customer has a system. These requirements will be functional in that they will describe what the system is to do or non functional in that they constrain some aspect of a system such as its response time.
- **Requirement Specification:** This is the process of detailing the properties of a system which have been discovered from requirements analysis. These are embodied in a document which is often called the system specification or requirement specification.
- **System Design:** This is the process of specifying the gross architecture of a system in terms of modules.

- **Detailed Design:** This is the process of specifying the individual modules in a system. It's helpful to add more structure to the service model stacks.
- **Acceptance Testing:** This is the final test of a system which carried out by the customer in the environment where the system will eventually be embedded. The acceptance test checks that the requirements specification has been correctly implemented. This is normally preceded by System Testing.
- **Modular Testing:** This is the process of checking out the individual modules of a system, the Modular testing sometime called as Unit testing.

4. PHASES OF SQA SYSTEM DEVELOPMENT :

Software Development Lifecycle is a subset of the SQA System Development Lifecycle. Here in this paper, the waterfall model is chosen as Software Development Lifecycle model. It is as follows:

- **Requirements Analysis:** In this phase, software engineers gather customer requirements by defining them with the help of customer and domain experts. Most of the time, a developing software must have interfaces with existing hardware and software. Therefore the information about them helps to define the interface requirements. In this phase, software engineers must understand the nature of the program to be built. Therefore, understanding the information domain, system's required functions, behaviors, performance and interface are musts.
- **Design:** In this phase, software designer identifies the system inputs, outputs and processes. Processing algorithms, data structures, databases and software structures are also defined.
- **Coding:** In this phase, software engineers and programmers transform the design into a code by using a selected programming language. Code review, unit tests, unit integration tests is part of this phase.
- **System Testing:** In this phase, the system is tested as a whole and system integration is realized. Activities are performed by the testing team.
- **Installation and Conversion:** After customer approval, the software is installed to serve the customer. If the new software will be used to replace the existing software, a suitable conversion process must be performed to prevent the interruption in the organization's services.
- **Operation and Maintenance:** Operation phase begins after the installation and conversion is completed. Maintenance activities are performed during the normal operation period which generally continues a few years.

5. TIMING ACTIONS OF SQA METHODS :

- Conduct Management's Role in SQA System process is the first step to build an SQA system in an organization. Therefore this process starts at the very beginning of Pre-Project Phase and it never ends.

- Construct Infrastructure of the SQA System is the base process to establish an SQA system in an organization. It starts at the beginning of Pre-Project Phase and it ends at design phase.

- Do Standardization and Certification is important to show that the quality of the software products produced by the organization has the level of chosen standards. It starts in the middle of the Pre-Project phase after constructing the infrastructure of the SQA system and never ends.

- Perform Contract Review starts at the Proposal/Contract Phase after having an offer from the customer and ends at the end of this phase.

- Develop Project Management Plan, SDP and SQA Plan start at the beginning of Project Preparation Phase just after signing a contract with the customer and ends at the second half of the design phase.

- Perform Activities to Force and Coordinate the SQA System is necessary to support the SQA system at the organizational level. Therefore it starts at the Project Preparation Phase before entering Software Development Lifecycle phases and it never ends.

- Conduct Quality Assurance Audits is an organizational level process to control the departmental SQA activities. Therefore, it starts in the middle of Requirement Analysis Phase just before the project level SQA activity starts. It ends after the project is delivered to the customer.
- Applying Software Quality Metrics Program is an auxiliary process to collect some data about the project and to use them to control the software development process. Therefore, it starts in the middle of Requirement Analysis Phase just before the project level SQA activity starts. It ends after the project is delivered to the customer.

- Perform Activities to Support the SQA System is necessary to support the SQA system at department level. Therefore, it starts in the middle of Requirement Analysis Phase just before the project level SQA activity starts. It ends after the project is delivered to the customer.
- The Apply Risk Management Program is important to take precautions against any possible problems which can be occurred during the development Lifecycle, and can cause undesirable deviations from project schedule and budget.

6. S/W QUALITY ASPECTS :

- Correctness:
  – Accuracy, completeness of required output
  – Up to datedness, availability of the information
- Reliability
  – Maximum failure rate
- Efficiency
  – Resources needed to perform a software function
- Integrity
  – Software system security, access rights
- Usability
  – Ability to learn, perform required task
- Maintainability
  – The effort to identify and fix software failures
- Flexibility
  – Degree of adaptability (to new customers, tasks, etc)
- Testability
  – Support for testing (e.g. Log files, automatic diagnostics, etc)

7. CONCLUSION :

Here, in this study the Software Quality Assurance concepts are being discussed that are used to make the error-free Software and concentrate on complex activities and used to complete in time and in cost estimation is prevented. As discussed earlier stage of this paper the Software quality Assurance activities, principles, factors and its methods are implemented in the early stages of software engineering development phases, because of this activity the software developer get the knowledge about the software what he is going to develop, it may reduce the rework and failures of the software's.

8. REFERENCES :

[1] Galin, Daniel (2004), Software Quality Assurance – From theory to implementation, Pearson –Addison Wesley, England.
[2] Omer Korkmazsystem, Simulation for software Quality Assurance (SQA).
[3] Darrel Ince, ISO 9001 and Software Quality assurance
[4] Sametinger, Johannes (2006), Software Engineering with Reusable Components, Springer Verlag GmbH; 1st Edition, ISBN-13: 978-3540626954..
[5] B. W. Boehm. Software Engineering Economics. Prentice Hall, 1981.
[6] Felipe Ortega, Daniel Izquierdo, Pedro Coca, Introduction to software quality evaluation
[7] ESA Board for Software Standardization and Control , Guide to software quality assurance
[8]The Software CrisisSource:http://www.unt.edu/benchmarks/archives/1999/july99 /crisis.htm)
[9] A Literal Review of Software Quality Assurance , International Journal of Computer Applications (0975 – 8887) Volume 78 – No.8, September 2013