Review Article [Dhanotia, 1(2): April, 2012]
ISSN: 2277-9655

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

Software Quality Assurance

Navin Dhanotia
Director of TechSuccessor PVT LTD, Indore (M.P.), India
navin.dhanotia09@gmail.com

Abstract

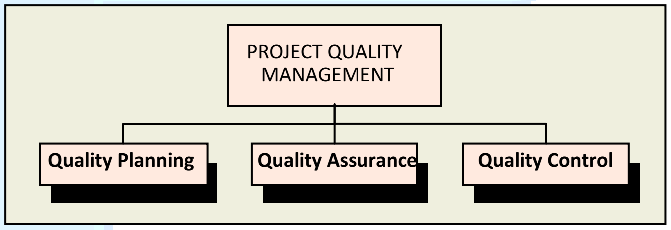
The Software Quality Assurance of any software application is quite simple, to makes sure the project will be completed based on the previously agreed specifications, standards and functionality required without defects and possible problems. In this paper i present the development process from the beginning of the project to ensure. If any software company develop software than quality assurance manager wants to see the maximum of software defects avoided at the earlier stages of the SDLC. So here we study about what's the main goal of Software Quality Assurance in our project or any kind of software.
One question is how can increase the quality of our software, so some points to be consider are as follows:-

- Understanding the types of defects escaping from the product.
- Using a model for the costs of finding and fixing defects using various quality assurance techniques.

Developing a quality process that included multiple inspections and test types

Introduction

In today's scenario software has taken a central role in our daily business and private life. Software is used in planes, trains, cars, banking systems and therefore the software's quality plays a crucial role. The quality is important for the acceptance of the software by the user and thus is a key factor to the success of the software product. Software systems are expensive products because their construction involves a lot of skilled people. Companies, which develop software often, spend excessive amounts of money to get high quality software, which overcomes the firms actual quality needs. On the other hand some companies do not take quality assurance seriously enough or do not spent enough money, or do not use the right techniques for the quality assurance of their software production. It has often been seen that companies let an immature software skip over to field production. A possible software failure may then lead to millions of breakdown costs, loss of reputation, and loss of market shares or even injure people. Thus, it is important to adopt the right way to achieve a good software quality assurance.
Quality assurance is the guarantee to maintain a certain level of quality according to target goals. Quality assurance is guided by a framework document that formalizes the quality assurance measures. The standard 8402-94 gives the following definition:-
"The series of reestablished and systematic activities laid out in the quality system framework that are performed when needed to prove that an entity will meet quality expectations."
With respect to software system quality, it is not always possible to achieve the "best quality", but the intension is to create a software system having the right quality for a given purpose.



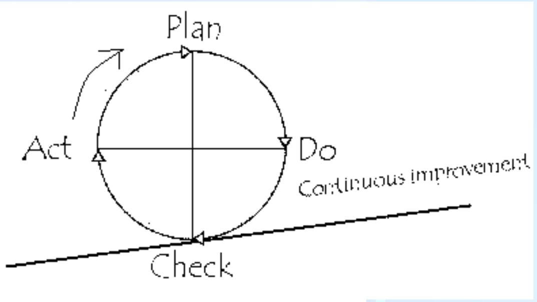
As a matter of fact, it is important for each software development project to define its specific meaning of software quality during the planning phase. On the one hand the quality of software is adequate if all the functional requirements are met. On the other hand the software's quality is also defined over non-functional requirements as reliability, usability, efficiency, maintainability and portability. This set of characteristics is defined in the international standard for the evaluation of software product quality ISO/IEC 9126-1:2001 [ISO01]. For each characteristic there exists a set of sub-characteristics which all contribute to the software quality to some extent. He states that quality requirements

Consist of three main parts, as shown in Table:-

Quality Requirement	Description
Condition	An optional condition that states under what conditions the quality requirements must be met. For example quality requirements specifying high performance and reliability may only hold during normal conditions, but not under degraded mode operations
Quality Criterion	A quality criterion is a system-specific description that provides evidence either for or against the existence of a given quality factor or sub factor.
Quality Threshold	A quality threshold specifies a minimum level of quality along a quality Measurement scale.

Continual Improvement

One of the basic principles of quality is prevention and continual improvement. This means that quality is a never-ending project whose goal is to spot dysfunction as quickly as possible after it occurs. Thus, quality can be represented by a cycle of corrective and preventative actions called a "Deming cycle":



This cycle, represented in the Deming cycle, is called the PDCA model. PDCA refers to the four following steps:

- "Plan: define the goals to be reached and plan how to implement the actions
- "Do: implement the corrective actions
- "Check: verify that the set goals are achieved
- "Act: depending on the results that occured in the previous step, take preventative measures

Why Do Software Projects Fail?

Some reasons are as follows:-

"What are the chances that project will succeed?"

For most companies and entrepreneurs starting on a new software project, this is one of the first questions that are considered. Unfortunately, the question is a very relevant one. Studies conducted by major research groups show that 50% of software projects fail today.

Why do software projects fail? Are there preventive measures to avoid software project failure?

A recent research project at 'The Standish Group' has identified reasons for software project failure. Lets have a look at the results of the research project. For purposes of the study, projects were classified into three resolution types – Resolution Type 1 or project success, Resolution Type 2 or project challenged and Resolution Type 3 or project impaired.

A. Resolution Type 1 or project success:
The software project is completed on time and on budget, with all features and functions as initially specified.
Among the Project Success Factors, it was noted that ' Clear Statement of Requirements' was ranked third in importance for the success of a project, after User Involvement and Executive management Support.

B. Resolution Type 2 or project challenged:
The software project is completed and operational but over-budget, over the time estimate, and offers fewer features and functions than originally specified.
'Incomplete Requirements and Specifications' was cited as the second main reason for exceeding the budget and time

C. Resolution Type 3 or project impaired:
The software project is canceled at some point during the development cycle. Here again, Incomplete Requirements was the main reason cited for the failure of the software project.

Overall, the success rate was only 16.2%. Challenged projects accounted for 52.7%, and impaired (canceled) for 31.1%.

The table below lists the criteria for software project success in order of importance.

SUCCESS CRITERIA	**POINTS**
1. User Involvement	19
2. Executive Management Support	16
3. Clear Statement of Requirements	15
4. Proper Planning	11
5. Realistic Expectations	10
6. Smaller Project Milestones	9
7. Competent Staff	8
8. Ownership	6
9. Clear Vision & Objectives	3
10. Hard-Working, Focused Staff	3
TOTAL	100

The chart clearly shows that 3 major factors greatly influence the success of a project - User Involvement, Executive Management Support and Clear Statement of Requirements. Out of these, "**Clear Statement of Requirements**" scores 15 points and is one among the top 3 reasons for the success or failure of a Software Project. It is clear therefore that Requirement Definition is a crucial step in Software Development.

Future Expansion

- The role of SQA with respect to process is to ensure that planned processes are appropriate and later implemented according to plan, and that relevant measurement processes are provided to the appropriate organization.
- The SQA plan defines the means that will be used to ensure that software developed for a specific product satisfies the user's requirements and is of the highest quality possible within project constraints.
- In order to do so, it must first ensure that the quality target is clearly defined and understood
- It must consider management, development, and maintenance plans for the software.

References

J. Woodruff Integrated Computer Control Software QA Engineer.

[Fenton 1995]
Fenton, Norman E. & Whitty, Robin. ―Introduction,‖ 1-19. *Software Quality Assurance and Measurement,*
A Worldwide Perspective, Norman Fenton, Robin Whitty, & Yoshinori Iizuka, ed.London: International Thomson Computer Press, 1995.

[Fowler 1990]
Fowler, Priscilla, & Rifkin, S. *Software Engineering Process Group Guide* (CMU/SEI-90-TR-024, ADA235784). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1990. http://www.sei.cmu.edu/publications/documents/90.reports/90.tr.024.html.

[IEEE-STD-610]
IEEE Standard Glossary of Software Engineering Terminology.

[ISO 2004]
International Organization for Standardization. *ISO 9000*. 2004.
http://www.iso.org/iso/en/iso9000-14000

[ISO 2004]
International Organization for Standardization. *ISO 9000*. 2004.
http://www.iso.org/iso/en/iso9000-14000

ISO 9001:2000. Quality Management Systems – Requirements. International Standards Organization, 2000, available at http://www.iso.org