

# Role of Software Quality Assurance in Software Development

Deepak Kumar<sup>1</sup>, Parul<sup>2</sup>

<sup>1</sup> Assistant Professor,  
Directorate of Distance Education,  
Kurukshetra University,  
Kurukshetra- 136119, India  
*indian\_deepakkumar@yahoo.com*

<sup>2</sup> Assistant Professor,  
SBMIMSAR, Baba Mastnath University,  
Rohtak- 124001, India  
*pau510@live.com*

**Abstract:** *Software industry is one of the leading one among other industries. The scope of a software application is increasing day by day and also the dependency over the software application is increasing. We can see software usage in medical laboratories, pharmaceutical laboratories, maintaining database of different organization, facilitating the sports person by visualizing their performance etc. So it is the prime responsibility of software industry to provide efficient and durable facility to its customers. This is a well known fact that every organization makes its mind earlier to achieve profit in their business and same here in the software industries. This paper represents the various aspects of the scope of the software application by impacting on the software quality.*

**Keywords:** *Software quality, Efficiency & Durability, Software scope.*

## 1. Introduction

Software quality is an important factor in software industry. The quality of software is judge by the customer satisfaction which can be achieved through applying standards. In the era of technology, every organization is emulating with each other for making this business optimal and with the adequate help of technology they are enabling themselves to stand in the field of competition. People and business rely on automated systems to deliver information quickly and efficiently through the digital processing of complex tasks. As business requirements become increasingly intricate, systems begin to intersect and overlap themselves, increasing the systems' complexity through the introduction of new factors and information. Now more than ever we are moving towards an era of intelligence and make connections that were previously unattainable. This level of computing is only sustainable through commitment to Software Quality Assurance (SQA). Organizations rely on developers to monitor and scrutinize the software engineering process and methods used throughout the Software Development Life Cycle (SDLC) to develop their systems and assist them in day to day operation of their business.

## 2. Methodology

The main role of the SQA (Software Quality Assurance) is to maintain the quality of the software products. For that it is to make sure that the standards and procedures are properly followed. Software Quality Assurance standards are developed to help organizations to achieve quality products. The standards and procedures include CMM1 and ISO but it is difficult and costly for small software development organizations to follow the standards [3].

Customers expect software quality to be very high, and yet we frequently hear in the news about bugs or design errors that cost time, money or worse – lives. As software development becomes more complex, innovative, and created by distributed teams, delivering quality is increasingly difficult. Quality assurance (QA) generally focuses on testing the software in the final stages of a project, when budgets are scarce and the pressure to deliver the product is high. To ensure quality, project managers must balance quality assurance with time-to-market.

Every application or business domain faces a specific set of software quality issues and software quality must be defined accordingly. It is important for each software development project to define its specific meaning of software quality during the planning phase. Such a definition contributes to the basis for setting objectives and practical measures of quality progress and determination of readiness for release to customers [1].

The customer sees what appears to be a working version of the software, unaware that the prototype is held together “with chewing gum and baling wire,” unaware that in the rush to get it working no one has considered overall software quality or long-term maintainability. When informed that the product must be rebuilt so that high levels of quality can be maintained, the customer cries foul and demands that “a few fixes” be applied to make the prototype a working product. Too often, software development management relents. [10]

## 3. Quality Assurance

*Quality assurance* consists of the auditing and reporting functions of management. The goal of quality assurance is to provide management with the data necessary to be informed about product quality, thereby gaining insight and

confidence that product quality is meeting its goals. Of course, if the data provided through quality assurance identify problems, it is management's responsibility to address the problems and apply the necessary resources to resolve quality issues. [11]

#### 4. Testing for Quality

Software engineers have a motto: "There is always one more bug." The amount of testing needed to ensure quality is hard to quantify. Very few project teams are able – or are willing – to predict how many bugs there will be in a release. Testing and fixing until all bugs have been found would be a very expensive task, if it were possible. And is it reasonable? Most organizations prefer to deliver a product on time, with a few unimportant defects, rather than to deliver it six months late. Needless to say, each industry has a different level of acceptable quality, for instance, word processors versus aircraft systems.

#### What part of your process needs to be improved?

Many organizations often limit QA to testing the final software. Functional testing may be outsourced to low-cost countries, which makes ensuring quality "somebody else's problem." In fact, this activity is quality control – checking after the fact that quality levels are met. On its own, quality control doesn't help teams build better software, a fact that can be very frustrating for the QA manager, as he is confined to the role of inspector of other people's work

In order to provide correct budget and schedule estimates, QA managers need the means to predict fault levels. Analyst reports indicate that the cost of fixing a defect increases dramatically the later it is found in the lifecycle – the cost of correcting a fault detected during final system testing can be up to 200 times more than if it is found during in the requirements phase. Very few organizations can confidently pinpoint the weakest links of their process, those where their budget would have the most impact. Without objective process performance indicators, QA managers cannot focus quality improvement initiatives where they are necessary or where they will have the best return on investment, let alone prove their efficiency.

Capability Maturity Model Integration (CMMI) was developed by the software Engineering Institute of CarnegieMellon University under the sponsorship of the US department of defense. Beginning with the Capability Maturity Model of Software (SW-CMM) and now continuing with the Capability Maturity Model International (CMMI) framework < software development organization have achieved significant gain in their ability to develop and deliver system with predictable results [6].

#### 5. Prediction of Defects & Errors

Defects and errors can be introduced and found during the different phases of the Software Development Lifecycle (SDLC). The practices described in this paper are easily applied to all types of processes (waterfall, iterative...) with different phase names. In this document, we will consider the following phases:

- Requirements analysis
- Design

- Implementation / Coding
- Unit Testing
- Integration testing
- System testing
- Customer usage (maintenance)

Team formation for the requirements gathering is also very critical. Mostly there is no SQA

representator in the requirements team. This leads to lack of domain knowledge and functional requirements. Compromise on quality is another problem area for lack of software quality in developing countries due to lesser profit margins. Most of the time project managers ask their QA team members to concentrate more on those projects where they can earn more profit. Profit margins are high for the foreign projects and foreign client demands very high quality. So team managers spend most of the time on foreign projects because they don't want to lose those clients. This approach affects the quality of the local projects.

#### 6. Inspection Benefits

Testing is widely employed by industry and formal verification widely advocated by the research community as methods of improving software quality. Inspection falls somewhere in between testing and formal verification. Formal verification has yet to catch on with software practitioners while inspection in one form or another has been widely adopted by industry and advocated by leading software practitioners. The main reasons for this difference in acceptance are that inspection can be used directly on the code itself, not abstract models of it, and because it does not require as substantial an investment in training as verification would require. It also does not require the time and formula manipulation ability that verification of typical programs would require. [7]

#### 7. Strategies for handling the risks in the software development process

Every software organization must prepare for the different kind of risks which may or may not occur in the software development process. The SRM- Software Risk Management should prepare the documentation of the previously handled or resolved risks in various phases so that they can be again helpful if that kind of risk arises in the software development. The risk monitoring process is done in such a way that the exact location of the software risk is defined easily and the identification of the software risk become easy. The resolution of the risk is totally depends on the reporting of the risk. If risk is minute or small in nature then it can be handled at team level or module level. If risk is potentially very high then the risk management is made aware by the different team leaders to highlight the issue within the time so that the risk management can take some necessary and immediate steps to recover those risks. After the resolution of the risks the risk management should note down all the activities which are used in the risk mitigation.

#### Conclusion

Software organizations can only get a respectable place in the world only if they keep themselves busy in maintaining the software quality. Here Software Quality Assurance plays an important role in making the business of Software

Company. Software testing is the activity that executes software with an intention of finding errors in it. Testing should be performed at different levels. Quality is the main focus of any software engineering project.

## References

- [1] Agarwal, R. ; Nayak, P. ; Malarvizhi, M. ; Suresh, P. ; Modi, N. "Virtual Quality Assurance Facilitation Model" Global Software Engineering, 2007. ICGSE 2007. Second IEEE International Conference on 27-30 Aug. 2007.
- [2] Jani, Hajar Mat "Applying Case-Based Reasoning to software requirements specifications quality analysis system" Software Engineering and Data Mining (SEDM), 2010 2nd International Conference on 23-25 June 2010.
- [3] Hansson, J ; Lewis, B ; Hugues, J ; Wrage, L ; Feiler, P ; Morley, J "Model-Based Verification of Security and Non-Functional Behavior using AADL" Security & Privacy, IEEE on 30 October 2009.
- [4] Lazic L., Kolasinac A., Avdic D., "The Software Quality Economics Model for Software Project Optimization", WSEAS Transaction on Computers, Issue 1, Vol. 8, January 2009.
- [6] Carnegie Mellon University Software Engineering Institute, CMMI overview Pittsburgh, USA, 2005.

[7] Parnas L. David, Mark Lawford ; The Role of Inspection in Software Quality Assurance, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 29, NO. 8, AUGUST 2003

[8] Pressman S. Roger; Software Engineering: A practitioner's approach Fifth Edition; ISBN 0-07-365578-3, Published by McGraw-Hill Page No. 32

[9] Pressman S. Roger; Software Engineering: A practitioner's approach Fifth Edition; ISBN 0-07-365578-3, Published by McGraw-Hill Page No. 224

## Author Profile

**Deepak Kumar** received MCA degree from Kurukshetra University, Kurukshetra in 2011 and UGC NET (Computer Science) in June 2012. In 2008, he is Gold medalist in PG Diploma in Internet Journalism of KUK. He has published 8 papers in the journals of national and international repute.

**Parul** received degree in Bachelors of Technology from Satpriya Institute of Engineering and Technology, Rohtak and M.Tech degree from Shri Baba Mastnath Engineering College, Rohtak. She has published 4 papers in the journal of national and international repute.