

## CHALLENGES OF SOFTWARE QUALITY ASSURANCE AND TESTING

**MD.SHAHADAT HOSSAIN**

Department of Software Engineering, School of Engineering & Computer Science,  
Independent University Bangladesh (IUB)

[shahadat.se@gmail.com](mailto:shahadat.se@gmail.com)

Mobile: +8801819825935

### ABSTRACT

Uncertainty exists in Software Company over the world. Software quality problem is leading issue for the software industry. The issue exists from 40 years or 50 years long. The industry is suffering and closing for this issue. In this circumstance it is important to address and remove its root cause. Otherwise, day by day industry economic loss will increase. I figure out some vital challenges of software quality assurance and testing which have been facing by software industries. The research focused on several small and medium software companies of the world. This paper represents different category of challenges along with responsible stakeholders. This research finds out that testing tools are available testing elements are available testing process has improved but still software has some testing challenges. My research figured out the bottleneck of challenges and explained in this paper. Here software engineers have scope to improve & overcome those challenges. This paper suggest systematic approach to solve the problem.

**Keywords:** Software Quality Assurance (SQA), Testing, Resources, Challenges, Software Firm, Programmer.

### INTRODUCTION

Software permeates our daily life. Software failures can lead to serious consequences in safety-critical systems as well as in normal business. For example "Ashton Tate", once the third largest company in the software industry, no longer exists (Humphrey, 2001) because of software quality defects. Quality is ensured by testing. Testing is an expensive process used to validate new software, important means of quality management and is widely used in industrial practice. It is estimated that 80% of software development cost is spent on detecting and fixing defects (ETH Zurich, 2010). To tackle this issue a number of tools and testing techniques have been developed to improve the testing. This research challenges of software quality assurance and testing has mainly focused on procedural challenges programming challenges & managerial challenges. This research started indicating from initial level weakened phases where defect generated first

to the final level phase. This research highlighted software engineering body of knowledge to overcome SQA challenges. It also shows the cost of bug fixing if it is not early detected and protected. The cost of testing accumulates from the time and resources spent on running the tests. To calculate the cost of a particular test here I provided a table of data. The goal is to explore whether the cost of testing or test process face challenges that should be solved in order to improve the cost-effectiveness of testing at the target company. The target company is a software developer or software implementer or in house developer whose main deliverable is the software product.

A recent report by the National Institute of Standards and Technology found that software failures cost the US economy about \$59.5 billion annually or about 0.6 percent of the gross domestic product according to a newly released study commissioned by the Department of Commerce's National Institute of Standards and Technology (NIST, 2002). At the national level over half of the costs are borne by software users and the remainder by software developers/vendors. The study also found that, although all errors cannot be removed, more than a third of these costs, or an estimated \$22.2 billion, could be eliminated by an improved testing infrastructure that enables earlier and more effective identification and removal of software defects. These are the savings associated with finding an increased percentage (but not 100 percent) of errors closer to the development stages in which they are introduced. Currently over half of all errors are not found until downstream in the development process or during post-sale software use. Software is error-ridden in part because of its growing complexity. Software developers already spend approximately 80 percent of development costs on identifying and correcting defects, and yet few products of any type other than software are shipped with such high levels of errors. Other factors contributing to quality problems include marketing strategies limited liability by software vendors and decreasing returns on testing and debugging according to the study (Ross, 2013). At the core of these issues is the difficulty in defining and measuring software quality. The increasing complexity of software along with a decreasing average product life expectancy has increased the economic costs of errors. The catastrophic impacts of some failures are well-known. But high-profile incidents are only the tip of a pervasive pattern that software developers and users agree is causing substantial economic losses.

Therefore, it is critical to improve software reliability to ensure long-term software operations without failures. Software testing have been the primary way for improving reliability. Software testing typically includes three steps generating test inputs, running test inputs and verifying actual outputs. However, with the ever-growing size and complexity of software systems software testing is becoming more and more challenging and costly. It costs billions of dollars and accounts for about 50% the cost of software development. In software testing, testers need to design a set of test cases, including test inputs and expected outputs, to cover most of the code and most bugs before releasing the software. It is challenging to generating such effective test cases manually for complex software systems nowadays. Automatic software testing reduces the laborious human effort in testing. However, the existing test input generation and test generation techniques often require manually provided specifications which may not be available. In this research I denoted such challenges of software quality assurance and testing faced by Software Company. Software testing is an integral

part of software development process. Although, during a testing process sometimes requirements have to be specified more exactly and sometimes requirements have to be analyzed. But this activity is not the body of testing, and it has to be done rather as a necessity. Management, despite that many organizations have test management positions. Certainly test engineers have to be controlled, but the testing itself is not management.

This study focuses on software quality assurance challenges, solution and practices, activities performed with software testing tools and also software testing standards of Software Company. Based on the outcomes of the study the contemporary practices of software quality in development domain are presented and also some recommendations regarding the best solution practices. Software testing is a Software Engineering knowledge area that has really matured from just being seen as an activity to being seen as a process closely inters woven with the complete Software Engineering process. Testing is no longer seen as an activity which starts only after the coding phase is complete, with the limited purpose of detecting failures. Software testing is now seen as an activity which should encompass the whole development and maintenance process and is itself an important part of the actual product construction. This description tells us that previously software testing was mostly done at the end of the development process (after all the code was written) and this approach is limited. It is stated that the development of a software product is mostly driven by four parameters: resources, time, scope and quality. In many cases the parameters resources, time and scope have a fixed value. The goal of software testing is to assure the quality of the end product. Nonetheless, in the real world resources has been almost always limited. This means that to develop a high quality end-product the available resources should be used optimally. All development activities should be adequate, therefore a development company should decide during its test planning which testing activities should be performed to assure the quality of the end product. There are many types of test activities. Each type of test has its own place in the development process. This development process is divided into the following phases, planning, design, coding and documentation, testing/fixing and maintenance. Although testing/fixing is mentioned as a separate phase, it clearly focuses on how to integrate testing in all phases of development. The research undertaken presents new and improved software testing challenges strategies which could be applied to any type of organizational information systems. It is expected that it will provide considerable assistance to the system testers/ developers/ software Development Company. This paper presents the challenges on a study conducted by the Information Technology (IT) industry. This was conducted during February 2016 – December 2016.

### **CATEGORY OF CHALLENGES**

During my work in several local and international software companies I faced different categories of challenges that are reducing the growth of the software firm. Software companies are facing those challenges continuously. I accumulated all those categories of challenges into three main categories. Each of the above categories has multiple types of challenges. This type of Software Quality assurance and testing challenge directly or indirectly affect the overall productivity of Software Company.

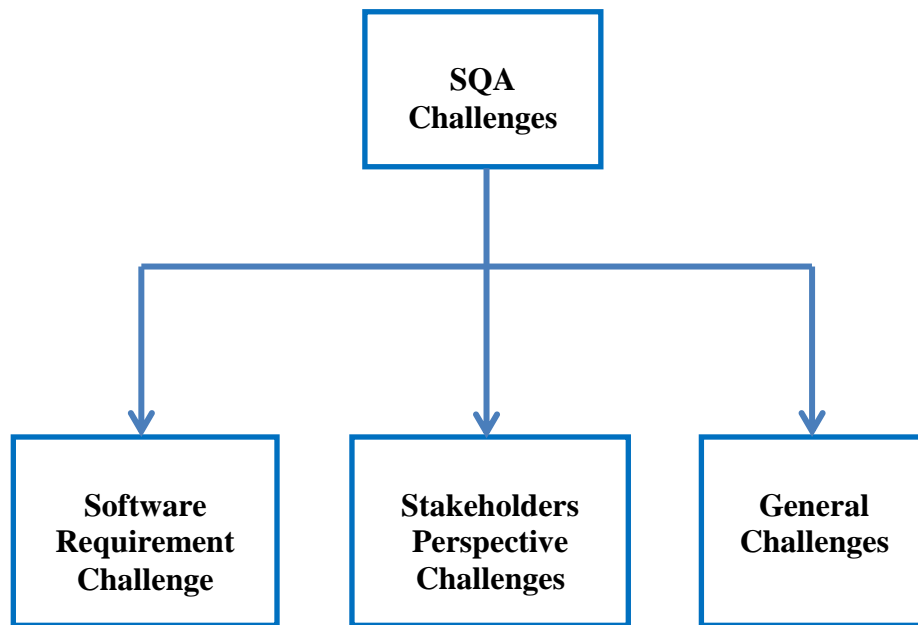


Figure-1: Category of challenges

Some challenges effect software company growth immediately some effects become visible after passing sometimes or after project implementation or post implementation and maintenance period.

### 1.1 Software Requirement Challenges

I said clients are very much conscious about software quality. But quality suffering initiates from software requirement collection. Improper way of requirement collection, poor & incomplete requirement, insufficient time for the requirement and lack of dedication makes a hole for quality failure. The initial challenge is collect right requirement at the right time (Teodoro, 2009). Sometimes software requirements collection engineers neglect to focus seriously on elicitation and validation of collected requirements. In some software firms they do not have separate software requirements collection engineer. Programmer or others person collect requirements. As a result requirements collects in unstructured way. This risk leads remaining development life cycle. Why people go to unstructured way for requirements collection and neglects the importance of requirements collection? It should not, it is unbelievable. My silent observation of the last 15 years figured out some carelessness in different point from different stockholders. That are not available in any book or report. In this research report I am mentioning this negligence's. In house software development face requirements challenges mysterious way. Requirements provider changes requirements randomly and submit one after one. As a result programmers fail to incorporate all requirements, in this gap logical & functional errors initiates that is one of the causes of software quality falls.

### 1.2. Requirements Collection Period

It is very vital period. If requirements collection remains incomplete, then it is affecting the rest of period. Sometimes project fails due to unclear requirements. I will explain it later, now telling how requirements avoid during requirements collection period. Software requirements collectors sometimes avoid the requirements if it is out of their

scope or the requirements is complex or time consuming. Sometimes unconsciously SRE avoid or forgot to elicit client requirements. The client does not provide requirement at a time. Client say start with this and show us a prototype if it became success then we will provide others requirements and changes where necessary. For this negligence's software quality suffer. Software requirements collector and requirements provider both are responsible for this quality suffering.

## **STAKEHOLDERS PERSPECTIVES CHALLENGES**

In this study I mainly focused on the question & tried to find out possible best causes & the answer of the question” why does software quality suffer” (Jeff, 2005)? Initially I found different types of stakeholders are responsible for software quality suffering. For example software requirement engineer and requirement provider. A very sensitive information is that in some cases user do not want to use software or they do not want to leave the legacy system for this they do not cooperate they provide wrong requirement so that software quality suffer and project fail. I distinguish stakeholder wise challenges into 4 types that are involved for this challenge's. My silent and close observation figured out some negligence of stakeholders about software quality assurance and testing. Any challenge either major or minor that affect software product is not ignorable by Software Company. The company should investigate and remove the challenge. Here are 4 types of key stakeholders responsible for these challenges.

### **2.1. Challenges From Developer/ Programmer Aspect**

There are lots of challenges from developer or programmer point of view on software quality assurance and testing. My analysis figure out that programmer is more intended to finish coding. They skip basic quality assurance some time they also skip unit testing too. I found their negligence & habit of showing over confident. They think it will work definitely so deliver it no need to test. But later programmer receive lot of bug reports. It is true programmer works under pressure and they just perform functional test then deliver. They do not give enough time for testing because they have multiple tasks in queue to complete. Based on situation programmer neglects software quality assurance and testing during development. Some programmers are very weak they do not have basic knowledge of SQA and SDLC. They do not realize the importance of SQA. Because they came from general education background such as political science, sociology etc. They just take a short training course of 3-6 months and learn the specific programming language such as dot net, PHP or oracle then start programming. They can't make proper programming logic as a result when end user input exceptional data then logical error occurs. Poor logical coding of programmer causes for software quality failure.

### **2.2. Risks From Company Point of View**

Software Companies are in a hurry to meet the deadline and they do not want to provide enough time for testing. They say if it works then deliver we will do the rest of testing during support and maintenance period or next release. Some of software companies neither have a SQA department nor a single quality assurance engineer. Private limited or SME who have in house development of software for their business they neither have SQA engineer nor think about SQA. They are willfully careless for software quality

assurance and testing.

### **2.3. Threat From Customer Side**

Clients focus on cost, they think more qualityfull software will cost more. Give us a software to perform our daily activities, we do not need high quality software. If it just meets our requirements then we are fine we are happy. These are real fact in developing country. Some clients willingly avoids quality assurance (Ricardo, 2007).

### **2.4. Negligence From Vendor Side**

Very exclusive and internal monitoring find out that due to vendor negligence software quality fail. Third party implementation vendor sometimes do not maintain software quality. They just make hurry to complete the implementation. They do some immediate solution to complete the implementation that is next create problem, and a lot of rework requires for overcoming this. Process mapping of existing software with current business process mismatch. As a result production environment does not work properly. I found a large project of Oracle E-Business Suit implementation vendor failed to go live with expected module within time and budget due to this challenge.

## **General Challenges**

These are common challenges according to software engineering body of knowledge. General level challenges are found in most of the firms that affect software quality assurance and testing. Following are general level challenges (Sommerville, 2013)

### **3.1. Economic Challenges**

The economic impacts on software quality assurance is high. I see due to quality issue user do not purchase software and for quality issues project become failure. Due to quality problem Software Company may shut down forever. So the impact is easily understood. Now we will see the individual impact or cost of quality control at each stage of software development life cycle SDLC. Cost of quality and bug fixing is varying from one stage to another stage (Rex, 2000). Initial stage of SDLC bug fixing cost is lower, later stage cost is higher. That's why I am focusing on early detection and correction of software. Following are cost of bug fixing in different stage of SDLC (Jones, 2009).

- To fix a problem at the requirements stage costs 1.
- To fix a problem at the development stage costs 10.
- To fix a problem at the testing stage costs 100
- To fix a problem at the production stage costs 1000.

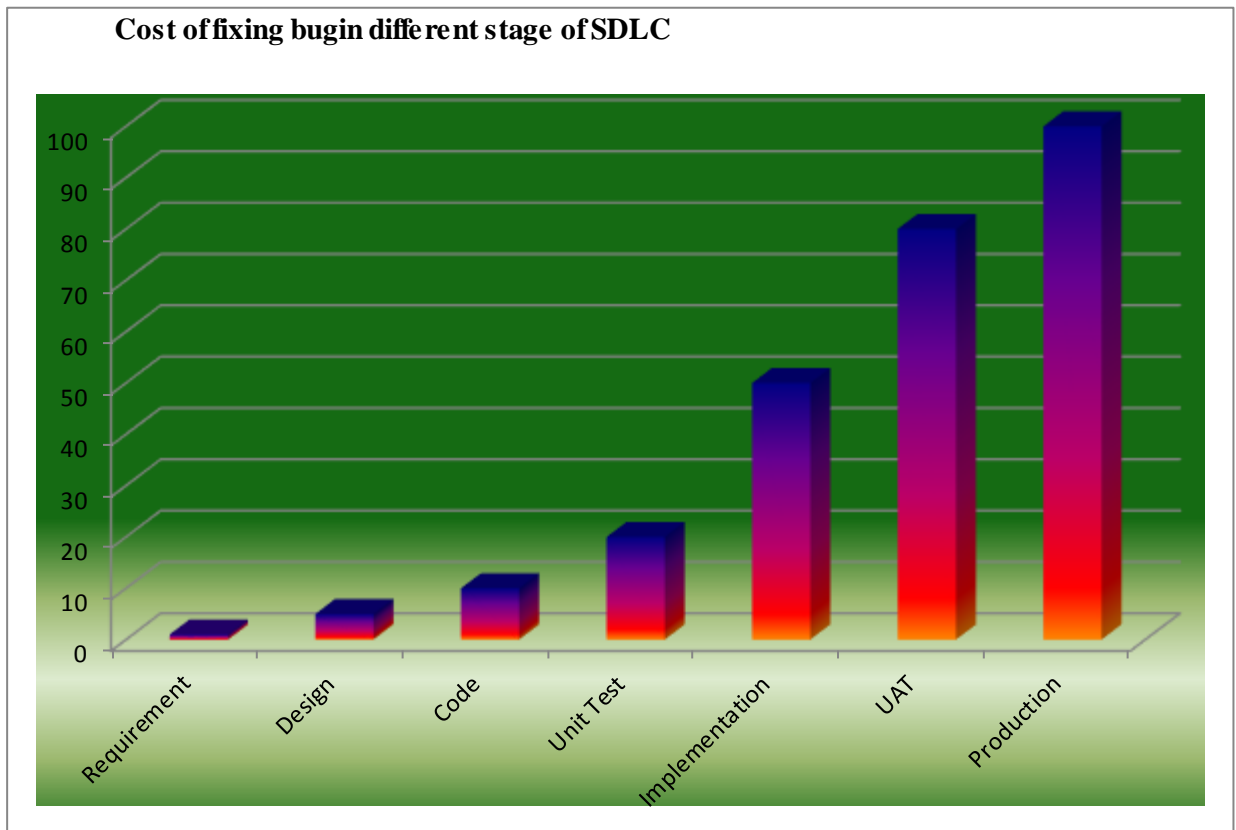


Figure- 2: Cost of fixing bug in SDLC

### 3.2. Resource Limitation Challenges

It is a common headache for more or less all software development firms worldwide. In some firms single person is completing total software development life cycle of a particular project. As a result one or more steps are missing consciously or unconsciously. Due to human resource limitation a single professional works on multiple project for multiple type of tasks either the professional expert or beginner in this field. This practice causes serious quality defect. Others resources limitation also directly or indirectly affect software quality for example hardware equipment, devices, testing tool and license product etc.

### 3.3. Environmental Challenges

Environmental challenge is a big issues for software quality assurance & testing as well as software development. Maximum time software development firms unable to provide an actual environment for development for testing. Software quality suffering challenge remains unavoidable for this limitation. I found environmental challenges in:

- A) **Hardware Environment:** Hardware environment is the vital factor for software development and testing. Primary server, secondary server, application and report server are differing from development environment to testing and testing environment to production environment. Insufficient hardware and others devices hamper development and testing process. Integration of hardware, machineries, device and others equipment that users are using and API that required to integrate with third party application as well as provide information for simulation is critical environmental challenges.

- B) **Working Environment:** Working environment is another vital issue for software development and testing. Congested and unhealthy office room, uncomfortable chair table and others furniture that is not only challenges for SQA but also makes serious disease for professionals. Small and medium entrepreneur software firms can't provide suitable working environment for development and for testing.
- C) **Test Environment:** Test environment required real life setting of target environment but for different type of limitation many times it is difficult to provide real environment of production for testing. Actual test case, test data and exceptional scenarios can't create by tester due to this limitation. Test engineer has limitation to write exceptional test case in advance that may occur few years later.
- D) **Freelancing Environment:** it is age of freelancing and virtual world. A laptop with internet connection is sufficient for work. Freelancer works here and there in that case development & testing environment is not getting focus.

### **3.4. Project Size Challenges**

Software development methodology also changes depending on project size. But the firm that only follows one process difficult to accomplish different projects. Size of the project is vital challenge. If the project size is too large then it is difficult to manage and maintain. Alternately if the size is too small then it gets lower importance. Sometime Software Company do not provide importance to small size or low cost project. Both the way quality suffers (Kshirasagar, 2008). Quality assurance for a multi-disciplinary project where different principle of different time zone work is very much difficult.

### **3.5. Number of Lines of Code Challenges**

The size of software products is no longer measured in thousands of lines of code but in millions (John, 2004). Whenever any unexpected change request come or error generate it is difficult to find out for accumulate and fix. A few years ago when processor and memory was lower compiler took a long time to compile a few lines of code program. Testing process suffer for multi million lines of code program. Web based application face more complexity to test line of code to fetch data from large database.

### **3.6. Technology Change Challenge**

Technology is changing upgrading every day. With technology changing if software can't change simultaneously in such case quality suffers. Many cases upper version software does not support lower version. Migration from legacy system to new system raise quality issue. Testing suffer for new technology if tester can't adapt to new technology rapidly.

### **3.7. Time Frame Challenges**

Real scenario is that the maximum project does not provide separate duration for quality assurance and testing. Testing is performed within development duration. Insufficient time for testing causes serious quality defect in the next phase I found user can't handover their requirements and real life transactional data within time frame. At last hour client submit their requirements that become difficult for incorporate in the software. Although requirements are incorporated into software by hard and soul try during day night but later quality problems arrive. Many software firms provide their best effort of development and testing at previous night before software delivery date or



project summation or demonstrations time.

### **3.8. Political and Legal changes**

I found government decision sometimes directly or indirectly take effect in software requirements, software development and quality assurance. Example are introduce new charges on particular items, digitalization of legacy system etc. External events lead to a change in government policies or regulations. In critical systems these may lead to requirements change Requirements engineering challenges. Many system requirements are influenced by the politics in an organization. Decisions on requirements are not made on a rational basis but are made because of the personal goals of stakeholder's requirements engineering challenges (Somerville, 2013).

### **WHERE DOES THE QUALITY SUFFER?**

The key question is why does software quality suffer & where does quality suffer? What is the initial phase software quality assurance should start? Many reasons are responsible for software quality suffering. A few of them are mentioned in example section. Software quality assurance should start from the very beginning to project closing like project planning, budgeting, and requirement collection. A survey by Testplant called Application Crisis Research found that 70 percent of businesses say that they are under pressure to innovate. About half say that they are releasing software without sufficient testing and 40 percent say that they are releasing software with no testing at all (John, 2017). Following are key points where software quality suffers

- Objective and scope
- Business value, benefit/cost
- Estimation
- Requirements
- Design, Coding
- Testing
- Integration
- Implementation
- Training
- Change management

### **Following Are Software Quality Suffering Example**

- Incomplete or erroneous specifications.
- Misinterpretation of customer requirements.
- Deviation from specifications.
- Violation of programming standards.
- Inconsistent component interfaces.
- Algorithms are weak

Following are SQA challenges metrics based on 10 software firms data

		F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
S R C	SRC	1	1	1	1	1	1	1	1	1	1
Stakeholder challenges	CDP	0	1	1	1	1	1	1	1	1	1
	CCA	1	0	1	1	1	1	1	1	1	1
	CVS	1	1	1	0	1	1	1	1	1	1
	CCS	1	1	1	1	0	0	1	1	1	1
General Challenge	ENC	1	1	1	1	1	1	0	1	1	1
	RLC	1	1	1		1	1	0	1	1	1
	ECC	1	1	1	1	1	1	0	1	1	1
	PSC	1	1	1	1	0	1	1	1	1	1
	NLC	1	1	1	0	1	1	1	1	1	1
	TCC	1	1	0	1	1	1	1	1	1	1
	TFC	0	1	1	1	1	1	1	1	1	1

Figure-3: SQL Challenges Matrix

**Note:** Here F1 to F10 are 10 Software Firms.

Data were collected from those firms for this research. There are maximum local software development firms of Bangladesh & a few international firms. This paper do not mentioning firms name in respect to privacy policy. This metric Shows a ‘D’ shape curve. All firms have above mentioned challenges except 1 or 2 challenges.

- SRC = Software Requirement Challenges
- CDP = Challenges of Developer/Programmer
- CCA = Challenges of Company Aspect
- CCS = Challenges from Customer Side
- CVS = Challenges from Vendor Side
- ECC = Economic Challenges
- RLC = Resources Limitation Challenges
- ENC = Environmental Challenges
- PSC = Project Size Challenges
- NLC = Number of Line of Code Challenges
- TCC = Technological Change Challenges
- TFC = Time Frame Challenges
- 1 = Yes, Firm has the problem
- 0 = No, Firm don’t have the problem

## **RESULTS**

The result of this research are above mentioned challenges that are true and available in software firms. My proposed probable solution to overcome above challenges is to prevent the challenge before it initiate. Protection because prevention is better than cure. This paper suggesting for focusing on software quality. Highly appreciates for providing quality is the top priority. Responsible stakeholders have to give enough time on each and every step of SDLC. Ensure the software quality assurance started from very beginning of SDLC such as requirement collection, requirement elicitation, documentation. Provide necessary training on new technology. Mindset that quality is the best policy. Utilities existing resources and try to overcome real environmental limitation for proper development and testing. Re-engineer the current business process & software development process.

## **CONCLUSION**

In this paper I have tried to create awareness on SQA challenges, financial loss by quality failure, and impact on firm growth and offer solutions to overcome those challenges. This research identified some major challenges that are facing by software companies. Similarly, this paper provided proposed solution to overcome that challenges. All of those challenges directly or indirectly related to software quality. I found that Software industry has 70+ years of working history. Most of the companies failed to show their success records. Some company succeeded, but their success rates were not continuing from beginning to date. Some companies started with big projects and made handsome money. Later they did not have such big project rest of the year they run business with the loss. Some company has multiple big project several times, but sometimes they don't have project at all. Some of them have enough revenue, but their profit didn't increase every years sequentially. A few companies shut down permanently with bank draft history. Some company sold out to another company. These scenarios are not only happening in the software industry, but also in others industry too. What are the reasons behind this scenario? Why Software Company failed to increase ROI successively? These are now the main concern of the software industry. This is multi-million dollars research issue. Currently my time and budget is limited that is why I can't expand the research. This paper figure out the root cause problems and provides solution from the software engineering body of knowledge. If Software Company can ensure the quality, maximum challenges will be minimized. The main aim of this paper is to help software industry for their continued progress in the software business. Increase their awareness to engineering the current business process where quality is falling. Here software engineering has scope to work to improve the productivity, quality and reliability of software products.

This research identified that ultimate loss is facing by software industry due to software quality issue but all related stakeholders are responsible for quality failure. Stakeholders can't avoid the responsibility because they are part of the company and when company become profitable then stakeholders are sharing the profit among them. This paper mentioned responsible stakeholders with authentic scenarios. This paper expects that all stakeholder's awareness will increase and they will be serious for software quality assurance and they should give top most priority on software quality assurance for industry profit for stakeholder's profit.

## **ACKNOWLEDGEMENTS**

I was inspired for this research by Dr.Rokonuzzaman and Professor Subrata Kumar of Independent University Bangladesh. I am grateful to my department for their cooperation. I would like to express my thanks to all software firm for their help. My wife Mrs Hosneara Shahadat and my Mother encourage me cooperated me to perform the research.

## **REFERENCES**

- ETH Zurich, Automated object-oriented software Testing using genetic algorithms and Static-analysis (2010). Swiss federal institute of technology Zurich 1, 1-3.
- Humphrey, Winning with Software An Executive Strategy (2001). Watts S. Humphrey. 1, 1-19. Carnegie Mellon University Software Engineering Institute. Print ISBN-10: 0-201-77639-1 Web ISBN-10: 0-321-57935-6
- Jeff, Software quality engineering-testing, quality assurance, and quantifiable improvement, Jeff Tian department of computer science and engineering southern Methodist university Dallas, 7x (2005).
- John Managing Software Deliverables (2004): A Software Development Management Methodology Managing John W. Rittinghouse. ISBN: 1-55558-313-X.
- Jones A short history of the cost per defect metric (2012). 2-2. Capers Jones, President, Capers Jones & Associates LLC.
- Kshirasagar, Software testing and quality assurance- theory and practice, Kshirasagar Naik -department of electrical and computer engineering-university of waterloo (2008).
- NIST, The Economic Impacts of Inadequate Infrastructure for Software Testing (2002). National institute of standards and technology. RTI Project Number 7007.011. Final Planning Report 02-3.
- Rex, Investing in software testing: the cost of software quality (2000). Rex Black is President and Principal Consultant of RBCS, Inc. R. Black, Managing the Testing Process, Second Edition. Wiley, New York, 2002.
- Ricardo, Testability of dependency injection (2007) University of Amsterdam Faculty of science, master research software engineering.
- Ross, The secrets to high customer satisfaction (2013).
- Sommerville, Requirements engineering challenges, Ian Sommerville (2013). 3-22.
- Teodoro, Software reverse engineering education (2009). San Jose State University - United States.
- John, Software Quality Suffers as Businesses Hurriedly Attempt to Innovate (2017), CEO Testplant.