A REVIEW ON CHALLENGES IN SOFTWARE TESTING

DR.G.ANIL KUMAR
Professor in Computer Science & Engineering
SCIENT INSTITUTE OF TECHNOLOGY,
Hyderabad, Telangana,[INDIA].

ABSTRACT

Software quality problem is leading issue for the software industry.The industry is suffering and closing for this issue. In this circumstance it is important to address and remove its root cause. Due to this delay lots of pressure in product deployment. I figure out some vital challenges of software quality assurance and testing which have been facing by software industries. In this paper I 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. In this paper I figured out the challenges in software testing.Here software engineers have scope to improve & overcome those challenges.

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

I INTRODUCTION

Software Quality Assurance (SQA) is simply a way to assure quality in the software. It is the set of activities which ensure processes, procedures as well as standards suitable for the project and implemented correctly. Software Quality Assurance is a process which works parallel to development of a software.[1] It focuses on improving the process of development of software so that problems can be prevented before they become a major issue. Software Quality Assurance is a kind of an Umbrella activity that is applied throughout the software process.Software permeates our daily life. Software failures can lead to serious con-sequences in safety-critical systems as well as in normal business. For example" Ashton Tate", [2][3]once the third largest company in the software industry, no longer exists 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[4][5]. 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[6][7] 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.At the national level over half of the costs are borne by software users and the remainder by software developers/vendors. [8][9]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, [10]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.

II KEY CHALLENGES IN SOFTWARE TESTING

2.1 Testing the complete application

It is quite impossible to test the complete application. There are many test combinations so it's impossible to test every combination. It will delay to ship the product, if you test all the combinations.

2.2 Relationship with developers

It is a big challenge. In this case, skilled testers can easily manage this situation positively. Developers or testers can give many justifications when they don't agree with each other for some points. For this, they requires good communication, troubleshooting and analyzing skill.

2.3 Regression testing

When your project expands, then the regression testing work turn out to be unmanageable. There may be so much stress to manage the changes in current functionality and previous working functionality checks and bug tracking.

2.4 Testing always under time constraint

When manager gives you an order to ship the project under a specific time then testers makes their main aim to simply complete the task and they do not focus on the quality of the product. There may be many tasks in terms of quality which needs to be completed within specified time.

2.5 Understanding the requirements

If testers are unable to understand the requirements properly then it will lead to bad testing because they will be unable to test the website or application properly. They requires good listening and understanding capabilities.

2.6 Decision to stop the testing

It is a very difficult decision. It requires skilled testers who can go through the testing processes and importance of each process and then can decide to stop the testing.

III CATEGORY OF CHALLENGES IN OF SOFTWARE QUALITY

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.

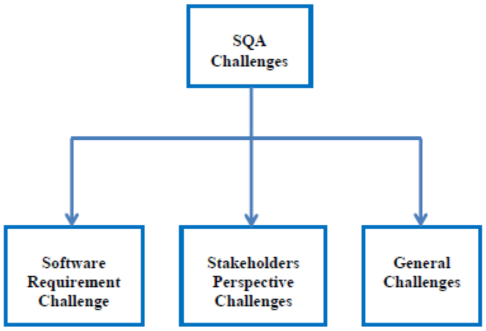
SQA Challenges
Software Requirement Challenge
Stakeholders Perspective Challenges
General Challenges

Fig 1: Hiearachy of SQA 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.

3.1Software Requirement Challenges

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.

3.2 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.

IV STAKEHOLDERS PERSEPECTIVES 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.

4.1 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.

4.2 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.

4.3 Threat From Customer Side

Clients focus on cost, they think more quality full 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).

4.4 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.

V WHERE DOES THE QUALITY SUFER?

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.Softwarequalityassurance[11][12][13] 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

VI BARRIERS TO GOOD TEST

Despite best efforts, and even in an agile environment, QA seems to get everything handed to them to test at the last minute or the very end of a sprint. What can be done at the eleventh hour to ensure a quality application?Well, while waiting for a testable deliverable, hopefully there should have been ample time to learn about the feature, read supporting documentation, and talk to the appropriate members of the product and development teams in order to come to an understanding of exactly what you need to do to test a feature. From this information, QA can develop a set of test cases to ensure whatever you need is in place prior to beginning test execution. Assuming the feature is relatively small, given that it was developed within a short sprint, you should be poised to execute. The time waiting for a feature to be delivered should be used to understand, define, and document your strategy for testing.

What about supporting documentation? Recently, I’ve seen more cases closer to no supporting documentation on the spectrum than to extremely detailed documentation. Most organizations require functional, nonfunctional, and technical specs, but even when those are all present, you cannot be sure that those documents contain everything you need to effectively test the feature. A good tester will go above and beyond what is given to them and seek out more information to fully grasp the feature.

VII 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.

REFERENCES

[1]ETH Zurich, Automated object-oriented software Testing using genetic algorithms and Static-analysis (2010). Swiss federal institute of technology Zurich 1, 1-3.

[2]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

[3]Jeff, Software quality engineering-testing, quality assurance, and quantifiable improvement, Jeff Tian department of computer science and engineering southern Methodist university Dallas, 7x (2005).

[4]John Managing Software Deliverables (2004): A Software Development Management Methodology Managing John W. Rittinghouse. ISBN: 1-55558-313-X.

[5]Jones A short history of the cost per defect metric (2012). 2-2. Capers Jones, President, Capers Jones & Associates LLC.

[6]Kshirasagar, Software testing and quality assurance- theory and practice, Kshirasagar Naik -department of electrical and computer engineering-university of waterloo (2008).

[7]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.

[8]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.

[9]Ricardo, Testability of dependency injection (2007) University of Amsterdam Faculty of science, master research software engineering.

[10]Ross, The secrets to high customer satisfaction (2013).

[11]Sommerville, Requirements engineering challenges, Ian Sommerville (2013). 3-22.

[12]Teodoro, Software reverse engineering education (2009). San Jose State University - United States.

[13]]John, Software Quality Suffers as Businesses Hurriedly Attempt to Innovate (2017), CEO Testplant.