Available online www.ejaet.com

European Journal of Advances in Engineering and Technology, 2021, 8(8):75-79

Research Article **ISSN: 2394 - 658X**

Tech World Without Quality Assurance

Bhupinder Paul Singh Sahni

bhupinder.sahni@gmail.com

ABSTRACT

Imagine a world where software applications, products, machines, etc. using software are getting developed, but there is no testing performed, no quality assurance in place to make sure that these software applications work as per business specifications, guidelines or safety measures. The world without Quality Assurance or Software Testing would be full of uncertainty, disorder and instability. Quality is one of the major factors in every project. In this paper, we will see the role of Quality Assurance in Software Development Lifecycle (SDLC). We will see some scenarios to imagine world without the quality assurance, and then having understood the importance of quality assurance, we will delve into how software testing or quality assurance is changing with the advances in technology and its future ahead.

Keywords: Quality Assurance; Software Testing; SDLC; Software Development Lifecycle.

INTRODUCTION

With the adoption of technology in almost every sphere of life, it becomes a great responsibility that different applications or software being used follow the quality standards. With the evolution of technology, software engineering has become more challenging and demanding. Technology or the Software being used should help out humans in their tasks, and not become a burden or a problem that could result in disastrous consequences. Software quality concerns are quite broad, including, for example, correctness, robustness, readability, and evolvability. A software failure occurs when a piece of software does not perform as required and expected. A software fault is a malformation whose execution causes a failure, and an error is a flaw in human reasoning and performance that leads to the creation of the fault. When examined closely, it is found that a software failure is a deviation of the execution of a program from its intended behavior [1]. Thus, having Software Quality Assurance or quality checks in place is important to make sure that the Software developed meets the desired requirements.

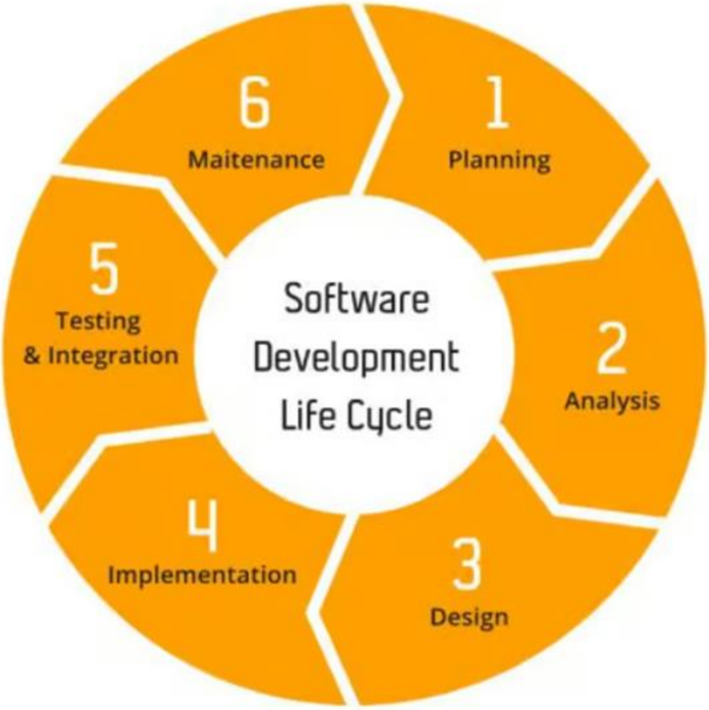
Software Quality Assurance (QA) stands as a critical pillar within the tech industry in ensuring the quality and reliability of software products before their release. Software Quality Assurance or Software Testing is the process of executing a program or system with the intent of finding errors. The objective of testing is to find problems and fix them to improve quality [2]. Software Testing makes our day-to-day experiences pleasant, giving us the sense of stability and connectivity. Quality assurance is essential for achieving customer satisfaction, ensuring product reliability, reducing costs, complying with standards, mitigating risks, safeguarding reputation, gaining competitive advantage, and fostering continuous improvement. It is a critical component of successful product development and business operations in the modern tech-driven world.

The intent of this paper is to showcase how important Quality Assurance plays a role in each phase of Software Development Life cycle, and therefore in the current technological world. Imagining a world without Quality Assurance would have its adverse consequences, which could also be catastrophic. We will investigate some of the disastrous incidents that have occurred in the recent past because of software glitches, that could have been avoided with proper quality assurance activities and quality checks in place. As we move into the current technological world, we will understand how Quality Assurance is getting upgraded with the ever-increasing demand for technology.

ROLE OF QUALITY ASSURANCE IN SOFTWARE DEVELOPMENT LIFE CYCLE

The Software Development Life Cycle or is also called as SDLC is a process used by Software Companies for developing Software. SDLC provides a structured process for managing the development process from gathering requirements to delivering the Software to Customers [3]. There are mainly 6 phases of Software Development Life Cycle. It has always been a big question of when to start testing. Experts suggest that every step taken in the development of the system must be tested thoroughly in a formal manner [2]. Quality Assurance (QA) thus plays an important role in each of the phases of SDLC to ensure that the final product meets predefined quality standards and requirements.

Let us study the 6 phases of Software Development Lifecycle and how QA plays a vital role in each phase of SDLC [4].



Planning: The first phase of SDLC is the Planning phase. In this stage, Requirements are mapped for the upcoming features. QA Analysts and Testers participate in this phase to give feedback on user experience and can point out potential issues. QA professionals collaborate with stakeholders to understand their needs and expectations, identify potential risks and ambiguities, and establish acceptance criteria for the software.

Requirement Analysis: In this phase, the Team discusses the problems or dependencies that team may have in execution of the Project. QA Analysts analyze the Tools to be used for the Project, different processes that can be followed, and review that the requirements are clear, complete, and align with the project goals.

Design: During the design phase, the development team converts software specifications into a design plan called the Design Specification. QA reviews design specification documents, and software architecture diagrams, which includes components, modules and interfaces to ensure that they align with the requirements and best practices. QA provides feedback on the feasibility, testability, and maintainability of the proposed design.

Development & Implementation: In this phase, developers write the actual code for the software product. QA may collaborate with developers to ensure that coding standards and best practices are followed. They may conduct code reviews to identify potential issues early in the development process. Parallelly, QA Analysts are also writing test cases for the functionality in this phase to be executed in Testing & Integration phase.

Testing & Integration: The testing and integration phase involves verifying that the software meets all of the requirements and functions as intended. The Integration of different applications is working as per requirements is verified. This is the most visible aspect of QA in the SDLC. QA analysts execute various types of tests, including unit tests, integration tests, system tests, functional test, performance test and acceptance tests, to verify that the software functions correctly and meets the specified requirements. This crucial phase of the software development life cycle uncovers the hidden defects in a software product. Regardless of the time-consuming and resource-hungry nature of testing, we can never ignore it [5]. Defects found in the system are also reported and tracked in a defect management tool. Regression Testing is also performed to make sure new feature development did not impact any of the existing functionalities.

Deployment & Maintenance: Once the software has been fully tested and approved, the product is finally ready for deployment. QA analysts are available during the deployment process to make sure deployment process is smooth and the software is properly configured in the production environment. As soon as the release goes live, QA does smoke testing of the Product to make sure that the deployment didn’t cause any issues and the product is stable.

After Software product is deployed, maintenance of the product happens. It involves ongoing maintenance and updates to the software to ensure that it remains functional and up-to-date. QA continues to monitor software’s performance in production and identify any issues that may arise.

Throughout the Software Development Life Cycle, QA analysts ensure that the software meets quality standards and that any defects are identified and addressed early in the development process. This helps to reduce the cost and effort of fixing issues later and ultimately leads to higher customer satisfaction.

A WORLD WITHOUT QUALITY ASSURANCE

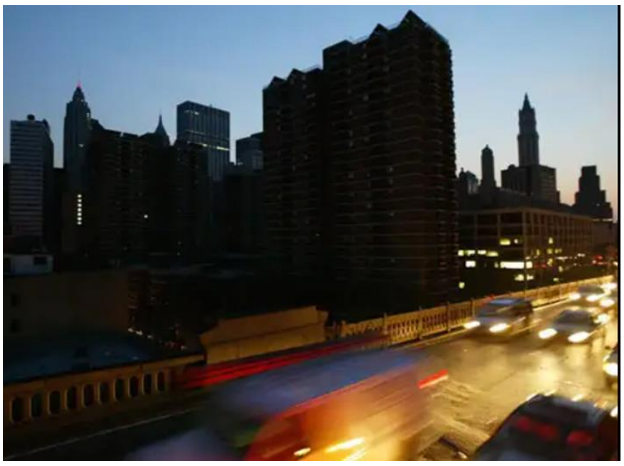
Quality is increasingly seen as critical to business success, customer satisfaction, and acceptance. Its absence may result in financial loss, dissatisfied users, and damage to the environment, and may even result in deaths.

For example, between 1985 and 1987, at least 5 patients were killed (and others critically injured) when a software-controlled radiation therapy system, the Therac-25, inadvertently administered massive overdoses of radiation — over 100 times the prescribed dose [6]. There have been incidents like these which have stressed that if Technology or software systems are a boon, they can also be bane if are not handled properly. In this case, it also stresses the importance of how important quality checks and quality control play an important role.

Another example is from August 2012 where Bugs in the Software coupled with lack of proper quality checks in place had disastrous consequence. A computer glitch nearly pushed investment firm Knight Capital into bankruptcy. Knight Capital was one of the most experienced companies in automated trading at the time. Knight Capital had a trade execution system, called SMARS (Smart Market Access Routing System), which was Knight capital’s algorithmic, high-speed order router. SMARS was able to execute thousands of orders per second and could compare prices between dozens of different trading venues within fractions of a second.

There had been substantial code refactoring in SMARS over the years without thorough regression testing. In the week before go-live, a Knight Capital engineer manually deployed the new code in SMARS to its eight servers. However, the engineer made a mistake and did not copy the new code to one of the servers. Knight did not have a second engineer review the deployment, and neither was there an automated system to alert anyone to the discrepancy. Knight also had no written procedures requiring a supervisory review. As a result of all these manual errors and without having quality assurance checks and processes in place, there was a glitch in the software which went Live. The firm lost about half a billion dollars in half an hour when a software error allowed computers to buy and sell millions of shares with no human oversight. It took only one defect in a trading algorithm for Knight Capital to lose $440 million in about 30 minutes. That $440 million is three times the company’s annual earnings [7]. The company's stock prices plunged by about 75 per cent in two days [8]. Knight Capital Group Holdings was eventually acquired by another market making rival, Virtu LLC, in July 2017 for $1.4 billion [9].

The 2003 North America blackout, at the time the second most widespread power outage in history, was the result of a local blackout that went undetected then cascaded out of control. General Electric Energy's monitoring software was affected by the same type of software problem that caused the X-ray disaster in the 1980s [10].



There are lot of disastrous incidents like above which really stresses the importance of Testing and having proper quality checks in place. These are extreme failures, but even small glitches in the software can lead to reputational, regulatory and financial losses to the company. In essence, quality assurance is essential for ensuring that technology products meet the expectations of users, adhere to industry standards, and operate reliably and securely. Without it, the tech world would be far less reliable, safe, and chaotic for users and businesses alike.

CHANGES IN QUALITY ASSURANCE AND FUTURE AHEAD

Today, testers need to test larger amounts of software in smaller periods of time. This makes it infeasible to simply execute entire test suites for every change. Also, it has become impractical to manually ensure that the tests cover all the changes. So, the focus of Quality Assurance has shifted to change-driven Testing and involving more Automation of repeatable tasks. Change-Driven Testing uses Test-Impact Analysis to automatically find the relevant tests for any given code change and Testers only those test cases. It increases the chance of catching mistakes early on, with a small amount of testing done. In addition to that, Automation Tests are run on a regular basis, so that any exiting functionality broken can be figured out and fixed. Automation Tests increase the speed of execution and thereby saving time for testers. But Automation Tests also have its disadvantages like insufficient reliability, need for support, fewer errors detected, and a false sense of product quality [11]. So, there needs to be a balance of what Tests should be Automated, and what portion of testing should still continue to be manual.

In addition to this, Artificial Intelligence (AI) and Machine Learning (ML) can also play a critical role in the evolution of testing. AI and ML algorithms can learn from the system to be tested and unearth potential issues, test complex patterns, find and prioritize bugs, perform root cause analysis, and much more with minimal human intervention. Also, as more and more applications are moving to Cloud, QA processes also need to ensure compatibility, security and performance across diverse environments [12].

Overall, the future of quality assurance will be characterized by greater automation, integration with development processes, a focus on user experience and security, and the adoption of emerging technologies like AI and cloud computing.

CONCLUSION

Successful implementation of Software projects is a critical strategic and competitive necessity for the technology today. So, Software Testing should be viewed as high-level risk management process, more like a parachute that is needed if You want to fly in the air without fear. In order to ensure this, having proper Software Quality assurance and quality checks in place is of paramount importance. The world without quality assurance in today's technological world is unimaginable, having already seen extreme consequences in the recent past. However, Quality Assurance would need to be modified to be more Automated, and change-driven; if it has to meet the current demand of complex applications being tested in lesser time.

REFERENCES

[1]. L. Osterweil, "Strategic Directions in Software Quality", ACM Computing Surveys (CSUR) 28.4, pp. 738-750, 1996.

[2]. M. Tuteja, G. Dubey, "A Research Study on importance of Testing and Quality Assurance in Software Development Life Cycle (SDLC) Models", International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, vol. 2, issue 3, pp. 251-257, July 2012.

[3]. S. Karunajeewa. "Software Development Life Cycle methods, their advantages and disadvantages", bootcamp.uxdesign.cc. Available at: https://bootcamp.uxdesign.cc/sdlc-methods-and-their-advantages-and-disadvantages-80095cea796c

[4]. "Software Development Models comparison", future-processing.com. Available at: https://www.future-processing.com/blog/software-development-models/

[5]. D. Kumar, K. Mishra, "The Impacts of Test Automation on Software's Cost, Quality and Time to Market", Procedia Computer Science, vol. 79, pp. 8-15, 2016.

[6]. K. Savoca. "When Software Kills", medium.com. Available at: https://medium.com/swlh/when-software-kills-ab6f48a15825#:~:text=A%20brief%20history%20of%20catastrophic,100%20times%20the%20prescribed%20dose.

[7]. M. Heusser. "Software Testing Lessons Learned from Knight Capital Fiasco", cio.com. Available at: https://www.cio.com/article/286790/software-testing-lessons-learned-from-knight-capital-fiasco.html

[8]. "Money: Knight Capital", newscientist.com. Available at: https://www.newscientist.com/gallery/software-bugs/#:~:text=Money%3A%20Knight%20Capital,shares%20with%20no%20human%20oversight.

[9]. H. Dolfing. "Case Study 4: The $440 Million Software Error at Knight Capital", henricodolfing.com. Available at: https://www.henricodolfing.com/2019/06/project-failure-case-study-knight-capital.html

[10]. "Infrastructure: The north-east US power outage", newscientist.com. Available at: https://www.newscientist.com/gallery/software-bugs/#:~:text=Money%3A%20Knight%20Capital,shares%20with%20no%20human%20oversight.

[11]. B. Oliinyk, X. Bai, V. Oleksiuk, "Automation in software testing, can we automate anything we want?", https://ceur-ws.org, vol. 2546, pp. 225-229, Nov 2019

[12]. J. Gao, X. Bai, W. Tsai, "Cloud Testing- Issues, Challenges, Needs and Practice," Software Engineering: An International Journal (SEIJ), vol. 1, no. 1, pp. 10-12, Sept 2011