Quality Assurance in the Digital Age: Exploring Contemporary Challenges in Software Testing

Humera Azam[1], Saim Masood Shaikh[2], Saad Akbar[3], Zamin Ali Khan[2], Mohammad Ayub Latif[3]

[1]Department of Computer Science, University of Karachi, Karachi, Pakistan
[2]Department of Computer Science, IQRA University, Karachi, Pakistan
[3]College of Computing and Information Sciences, Karachi Institute of Economics and Technology, Karachi, Pakistan

Abstract: Software Quality Assurance Engineers (SQA Engineers) assess and analyze the reliability of software systems and their components, and perform usability tests on software systems. SQA Engineers face unique challenges that may affect their efficiency, precision, and accuracy as they execute their testing functions. This paper aims to identify the challenges most SQA Engineers encounter as part of their daily professional activities. The paper first describes the phases of the software testing life cycle in detail. It discusses the relevance of SQA to software testing Automation. It focuses on the challenges SQA Engineers face in the current software market, 20, which negatively affect the overall performance and efficiency of SQA Engineers. This paper aspires to explain the reasons for the challenges SQA Engineers face 20, which inevitably lead to software delivery and the pressure of tight deadlines. For this purpose, a number of software development companies were studied to identify the varying and overlapping challenges they face in the performance of their roles. It SQA Engineers and assign the challenges of communication, task allocation, documentation, Quality Assurance (QA) automation, and the challenges of evolving technology. This illustrates the complexity and multi-layered quality of the challenges, which, in turn, may cause delivery of substandard software. It is, therefore, most important that SQA Engineers be allowed to complete their current SQA assignments. Although they expressed more general apprehension regarding the software landscape, knowing how to manage these issues can simplify software testing, thus speeding up deployment timelines considerably while also enhancing the quality of the software.

Keywords: Software Quality Assurance, Software Testing, Automation Testing, Quality Assurance, Requirements, Challenges
Email: humera.azam@uok.edu.pk

1. Introduction

Software Quality Assurance Engineers play a very important and critical role in ensuring the quality of software products by testing and validating them before release, because it is very important for client satisfaction and the working of software according to its requirements and function [1]. However, SQA Engineers face several challenges in the whole software testing process. These challenges include dealing with constantly changing requirements by the client and project cost, and sometimes these requirements are changed by the team lead according to managing the resources, identifying, investigating, and fixing bugs in complex software systems and processes, managing test data and environments, and ensuring compatibility with different platforms and devices [2]. Software Quality Assurance Engineers may need to navigate communication and collaboration challenges with cross-functional teams, manage time constraints and project deadlines, and stay up-to-date with evolving testing technologies and methodologies. Sometimes SQA Engineers have to face show-stoppers in testing due to critical bugs in the software [3]. Overcoming these challenges requires a combination of technical expertise, effective communication skills, and a strong commitment to quality.

According to Figure 1, Software Quality Assurance Engineers face challenges in testing resources, short testing time constraints, ambiguous requirements, lack of communication, and changing project priorities [4][5].

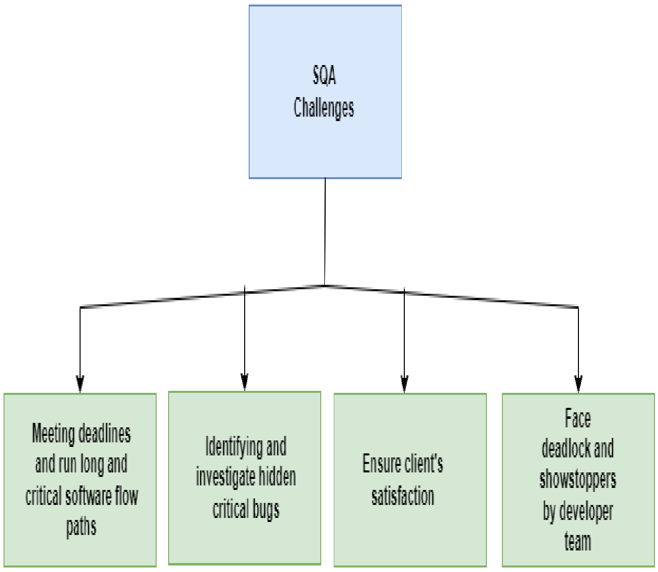

Figure 1 SQA Challenges

Many challenges are faced by Software Quality Assurance Engineers during software testing. Previous studies have proven that software testing is a very important and critical phase in the whole software development life cycle, ensuring the best quality of software systems [6]. Software Quality Assurance Engineers face several challenges in this domain that can impact the effectiveness and completeness of testing efforts. Following is the flow of this research paper.

Figure 2: Flow Chart of Paper

We will follow the flow according to Figure 2 for this research paper. In the Introduction, there is an overview of the whole research paper, including the need and importance of Software Quality Assurance in Software Engineering, the important role of Software Quality Assurance Engineers, and the need to address important challenges and find effective solutions. Then in the literature review, we have summarized all important key findings, methodologies used limitations in the existing literature. In the next section, we have mentioned the main and important challenges faced by Software Quality Assurance Engineers. Then we have proposed a proper solution to avoid the challenges in Software Quality Assurance. The last section of this research paper is the conclusion, in which we have summarized the importance of this research paper and key findings for the challenges faced by Software Quality Assurance Engineers in software testing.

Many challenges are faced by Software Quality Assurance Engineers during software testing. Previous studies have proven that software testing is a very important and critical phase in the whole software development life cycle, ensuring the best quality of software systems. Software Quality Assurance Engineers encounter several challenges in this domain that can impede their effectiveness and accuracy, ultimately affecting software quality and deployment timelines [7]. The most important challenge is the increasing complexity of software systems. We all know that software systems become more modern and complicated. Abdullah et al. say that to ensure the best output of the system, it is mandatory and important to test all functionalities, interactions, and important cases [8]. Software Quality Engineers must design and execute efficient test cases to ensure adequate coverage, which requires significant time and effort. Limited time and project deadlines are also a challenge for Software Quality Assurance Engineers in software testing. With this, there is pressure to complete testing within limited timeframes, which can result in insufficient test coverage. To overcome these challenges, companies need thorough testing with project timelines that are crucial to avoid compromising software quality [9]. One more study shows that communication is a big hurdle, further adding to the challenges of Software Quality Assurance in software testing, especially in faraway, geographically distributed teams. Effective coordination between team members, sharing information, and resolving issues can be difficult when working remotely or across different time zones. Myers, G. J., Sandler, C., & Badgett, T. say in their research that Software Quality Assurance is an important way to validate software requirements that are defined by the client, and it is used to save cost and production time. Software Quality Assurance is used to make efficient and important improvements in system infrastructure, and this can save a total third of the cost [10]. Azeem Uddin, in his research, described the objectives of software testing and addressed their challenges and as well as different levels of software testing to gain the maximum cost-effective software that can fulfill all requirements that are described by the client [11]. This literature review explores the contemporary challenges faced by SQA Engineers, drawing on recent research and scholarly discussions.

1.1 Importance of SQA Engineers in Software Testing

The function of SQA Engineers regarding software testing has been well addressed in the recent literature. The importance of their involvement in software quality and throughout the entire software development process is underscored by various scholars [12].

1.2 Challenges in Software Testing

Challenges during software testing faced by SQA Engineers are numerous. More recent research highlights specific issues with communication, balancing resources, record keeping, automating tests, and keeping up with changing technologies [13]; [14]

1.3 Impact on Software Development

These challenges and their impacts on software development have also been examined. Unresolved delays in testing and issues relating to quality can propagate project overruns and cost escalations, which incentivize organizations to deal with these challenges more proactively [15].

1.4 Root Causes and Responsible Parties:

Research has delved into the root causes of these challenges and the individuals or teams responsible for them. Understanding these factors is crucial for devising effective mitigation strategies [16]

1.5 Solutions and Mitigation Strategies

Contemporary literature provides insights into potential solutions and mitigation strategies for these challenges. These include improved communication channels, enhanced documentation practices, resource optimization, and the adoption of advanced testing tools and methodologies [17].

1.6 Industry-Wide Implications

The broader implications of these challenges on the software industry have also been explored. The challenges faced by SQA Engineers can impact the entire software ecosystem, making it imperative to address them at an industry level [18].

1.7 Call for Research and Action

Research and the proactive measures are described as "needed." Gaps require attention. Best practices need to be developed. There needs to be adaptation to the changing industry, documenting effective practices. It appears that ensuing literature and the SQA Engineer role most important for contemporary literature recognizes the SQA Engineer role most important. Should be "software testing." There needs to be improvement in the Q and deployment for "software testing." Timeliness of deployment should be enhanced. "R&D." "The software industry Q and Timeliness of deployment." The gaps are most important. "pro " "ag " and the value must be provided.

2. Problem Statement

Challenges in software testing and quality assurance stem from software quality, software functionality, and client stability and satisfaction. With regard to software testing challenges, the fragmentation and evolution of software developments on a continuous basis make complete coverage during testing a sparse prospect. Within software systems, the architecture is made up of finer and more complex systems, which makes debugging ever more complex, requiring systems thinking and adaptive problem-solving. Moreover, challenges in the scaffolding and orchestration of disparate systems and stakeholders over time, dependency axes, and the rapid advancement of tools and methodologies that aid in testing. In overlooking the challenges above, one is bound to fail in attaining quality software that can cause client dissatisfaction.

3. Main Challenges in Software Quality Assurance

Amidst the landscape of challenges faced by Software Quality Assurance Engineers in their pursuit of software testing excellence, it is crucial to delve deeper into the core obstacles that form the bedrock of their daily endeavors.

3.1 Testing the Whole Application in a Limited Time

Due to short project deadlines, it's a big challenge for Software Quality Assurance Engineers to test the whole software in a short time with efficiency. If they want to test all key functionalities of the software, then it causes delays in meeting deadlines [20].

3.2 Communication With Software Engineers

Dealing with this situation is a challenge for Software Quality Assurance Engineers. When Software Quality Assurance Engineers and Software Engineers present their different opinions on certain bugs or problems, they mostly present various justifications for a certain problem. Handling this challenge successfully in such situations needs strong communication skills, adept troubleshooting abilities, and proficient analytical capabilities [21].

3.3 Perform Testing Consistently Within a Limited Time

Most project deadlines are set by team leads or managers, and many Software Quality Assurance Engineers only focus on completing the testing within the time limit set by their team lead or manager. Software Quality Assurance Engineers miss the key points of software to test, which affects the quality of the software product [22].

3.4 Software Requirements Understanding

This is the problem that sometimes Software Quality Assurance Engineers are unable to understand the requirements of software, and this results in inefficient and low-quality testing of the software. Software Quality Assurance Engineers require good analytical and communication skills to understand the key functionality of software [23].

3.5 Testing the Regression of Software

With time, when the software evolves with new features, the regression testing gets more challenging. This creates stress on Software Quality Assurance Engineers to test the new functionality and previous functionality with proper bug tracking [24].

3.6 When to Finish the Testing

It is the most challenging and important decision when to finish the testing and deploy the software; it requires the most excellent analytical skills for the Software Quality Assurance

Engineers to determine when to stop the testing and ensure that the software will not produce bugs anymore [25].

4. Challenges Category in Software Quality Assurance

As a Software Quality Assurance Engineer, I have faced many challenges during my work tenure in a software company; these challenges are the main hurdle to the growth of a software company. Software companies face these challenges daily. I have categorized these challenges into the main 3 categories. These types of Software Quality Assurance indirectly and directly affect the growth of software companies and make it more challenging to gain market strength.

Some challenges affect the growth of software companies in a very catastrophic way, and some challenges occur after the deployment and implementation of software products on the client side.

4.1 The Dedicated Period for Requirements Gathering

The most important time in software development is requirements gathering; if requirements gathering remains incomplete, then it shows a catastrophic loss to the rest of the software development process. Unclear requirements fail the durability of the whole project, so it is very important that all requirements are clear and free from all types of ambiguity for software development. Sometimes those who are responsible for requirement gathering avoid the requirement gathering if the requirement gathering is out of scope or when the requirement is very long, time-consuming, and complex. Sometimes Software Requirement Engineer forgets the requirement that was added by the client. The other problem is that sometimes the client says that they will not give any requirements to make a prototype first; if the prototype is of the client’s expectation, then they will give other requirements. With these challenges, software quality suffers, and this is caused by the negligence of the Software Requirement Engineer and collector, which affects the quality of software products.

4.2 Challenges in Software Requirements

In the software industry, clients are very worried about the quality of software because they are paying for the software product, so their priority is always that the software must be bug-free and efficient in working. As we discussed above, challenges in software quality start to suffer from the requirement phase. Unprofessional way of requirements, short period, and lack of dedication make way for the failure of software quality. The first challenge is to collect the important requirements at the right time. Sometimes, Software Requirement Engineers show carelessness in the validation of collected software requirements. Many software companies do not have any Software Requirement Engineers; this work is done by

Software Quality Assurance Engineers and Software Engineers, and due to this, requirements are collected unprofessionally in a very unstructured way. This starts the risk in the whole software development cycle. I don't know why people are so careless in the requirement phase. My observation in a software company concluded with carelessness from different stakeholders. In this paper, I have mentioned some very serious negligence in software companies that is the main cause of challenges in software testing. Sometimes requirements are submitted to the Software Engineer one by one, and this results in the Software Engineer's failure to understand the main logic. This logical gap initiates a functional error that causes a big challenge for the Software Quality Assurance Engineer.

5. Challenges in Software Testing from the Perspective of Stakeholders

In this research, I have studied and focused on the main challenges of software quality assurance and testing. In this study, first I tried to find the answer to the best question, which is "Why does software quality impact and suffer?". In my study, I have found that many types of stakeholder are responsible for this, and due to their negligence, the software quality suffers. For example, if a Requirement Engineer does not gather requirements properly, then it will create trouble in the whole process of software development. A very sensitive topic is that sometimes the user does not want to give proper information to the Requirement Engineer and does not want to cooperate with them, so the software quality suffers, and the project is unable to complete the important requirements. I have categorized challenges according to stakeholder types into four types. My study has figured out some important and catastrophic negligence regarding software quality assurance and software testing Challenge is whether major or minor, it surely affects the software product. Every software company should work and think about these challenges to improve the quality of its software products. The four main responsible stakeholders are as under.

5.1 Risks According to the Perspective of the Company

As we all know, all software companies are always chasing short deadlines and put pressure on software engineers and software quality assurance engineers to meet the deadline as soon as possible. In this hurry, many software companies give a very short time for software testing, and that results in defective software delivery to clients. The company's perspective is that if the software is working, then deliver it to the client as soon as possible, because the rest they will do in the support and maintenance phase, and deliver bug-free software in the next release. Some software houses do not have any software quality assurance engineers, and they do not give importance to software testing. Small software houses have untrained quality assurance engineers who do not focus on software quality assurance.

5.2 Serious Negligence from Third Third-Party Vendor Side

According to my serious and keen monitoring, I have studied and discovered that software quality suffers and is challenging due to the serious negligence of third-party negligence. The software, which is implemented by a third party, does not properly maintain its software quality. They just provide immediate solutions in the implementation of the software, which results in defects in other modules and the evolution of software, and they are always in a hurry just to implement their software without maintaining the quality of the software. Due to their negligence process mapping of software for the current business mismatches and a lot of effort to overcome these problems. I have found in my study that a very large number of E-business projects failed in the live environment just because of the negligence of the vendor, and this creates a loss in the budget of software houses.

5.3 Potentials Threat from the Customer's Side

As we know, customers want the best quality software within a limited budget. Some customers want that they just need normal software for their daily activities and need and they do not want high-quality software. Their perspective is that if the software is working, then they are happy, and due to this, they avoid quality assurance and testing of the software.

5.4 Software Engineering Challenges

Software Engineers are always intending to complete the code just within the deadline, which is why they avoid completeness and quality in the coding. According to my research, they skip the basic rules of maintaining the quality of software. Due to their negligence, Software Quality Assurance Engineers have to face many troubles and challenges in software testing. I have noticed that some software developers skip unit testing because they are too overconfident that their code will work best, and they say that no need for testing, just deliver it. But later, there were a lot of bugs in the release that were found by Software Quality Assurance Engineers. There is another point and reality that software developers work under pressure, so they just test functional requirements and deliver them for software testing because they have multiple tasks to complete. Some software engineers are very weak in the basic process of SDLC and Software Quality Assurance; they don't know the importance of Software Quality Assurance. Some software developers are not well-trained they make logical errors, which is a challenge for Software Quality Assurance Engineers to produce correct results.

6. Efficient Software Testing Barriers

In the agile development method environment, the software quality assurance team handles every testing in the last minutes of delivery because the software is delivered to them very late; due to this, it is impossible to perform efficient testing in the few hours of software delivery [26]. In a short time, only some important functionalities can be tested; the rest minor bugs remain untested due to the short testing time. But one thing can be done for good testing and that is when waiting for the deliverable release, the software quality assurance engineers can read the documentation of software, talk with the appropriate member of the software, make query and assumption documents, and make necessary important test cases so that when testers get the release they can test everything important for the functional success of software product. When the module is small, it means it will require a small time for software testing. But when the module is big and complex, then during the waiting time for release, software testers make effective and fine strategies for excellent software testing.

The other problem that I have found in my study is that there is no supporting document concept in many software houses. Supporting Documents are extremely important in the SDLC. Many software houses have software requirement specification documents, scope documents, functional and non-functional documents, and technical specification documents, but we cannot be sure that all these documents have all the required information that a software quality assurance engineer needs in software testing to test all the required important features of software. An excellent software quality assurance engineer will always go beyond all information to test the most important functionality of the software and the success of the whole software project [7].

7. Testing in Agile Methodologies

During the development phase of software, there is a term we use, and that is ‘agile’. In this term, we usually consider collaboration, simplicity, and flexibility [9]. After this, we come to software quality assurance using agile methodologies and find out all defects, bugs, and errors in software [27]. There are also many challenges that are faced by Software Quality Assurance Engineers during testing in agile methodologies. These challenges are as under:

7.1 Documentation for Testing in Agile Methodology

Proper documentation is considered a major challenge for software development in the agile method. In the agile method, the client is continuously changing the requirements and keeps forcing for rapid delivery of software and neglecting the requirement of proper documentation. Due to the pressure of delivering software, the rapid programming method is used, and in rapid programming, it is a big challenge to make requirements documents, test cases, and execution of test cases [28].

7.2 Insufficiency in Coverage for Testing

Previously, we discussed that the client is continuously changing the requirements in software testing, and this causes instability in the project. When requirements change quickly, maintaining test coverage is very difficult due to continuous changes in the code [29].

7.3 Detection of Defects At An Early Stage

Defect detection is very hard in agile methodology. Defect detection in the early stages is very challenging, but it is done at a lower cost and with less damage. Software Quality Assurance Engineers are under pressure for the early detection of defects in software to save cost [30].

7.4 Code Break Due to Frequent Builds

When code changes daily due to changes in requirements, this breaks the code. Testing is required for the code break, and testing resources are not sufficient every time. At this time, automated testing can be very useful [31].

7.5 Lack of Focused Testing

As we have discussed above, we all know that there is no proper phase for software quality assurance and testing in agile methodology. The focus is on rapid programming, and in this phase, software engineers are developing rapidly. Due to all the rapid changes, focus testing is lacking because of not a proper phase of testing in agile methodology [32].

8. Statistics of Challenges Faced by Software Quality Assurance Engineers

The table outlines the types of challenges faced by Software Quality Assurance Engineers in software testing, along with their corresponding percentages based on the survey of 5 software houses in Karachi, Pakistan.

Table 1 SQA Challenges Statistics

Challenge Type	Percentage
Test Data Management	12%
Test Environment Setup	15%
Technology Advancements	10%
Collaboration and Communication	14%
Test Execution and Analysis	22%
Test Case Design	20%
Defect Tracking and Reporting	17%
Test Automation	28%
Lack of Requirements Clarity	25%
Time Constraints	18%

The above statistics show that 12% of Software Quality Assurance Engineers face challenges in test data management because obtaining and managing suitable test data is very important. However, 15% of Software Quality Assurance Engineers face difficulties in setting up complex test environments. Staying updated with the latest technology is also a challenge, which is faced by 10% of Software Quality Assurance Engineers. 14% of Software Quality Assurance Engineers face challenges in proper communication with software engineers, project managers, and stakeholders. Executing proper test cases and analyzing the results is a challenge that is faced by 22% of Software Quality Assurance Engineers. 20% of Software Quality Assurance Engineers face problems in creating effective test cases. Managing and tracking severe bugs throughout the testing process is a challenge faced by 17% of Software Quality Assurance Engineers. Creating automation scripts challenge faced by 28% of Software Quality Assurance Engineers. Incomplete and ambiguous requirements challenge faced by 25% of Software Quality Assurance Engineers. Complete testing in a limited time is the main challenge which is faced by 18% of Software Quality Assurance Engineers.

9. Analysis of Important Metrics for Assessing Challenges in Software Quality Assurance

Here are metrics that are used to calculate the challenges in Software Quality Assurance:

9.1 Requirement Understanding

Requirement coverage is used to measure the percentage of requirements that have been covered by test cases. It is calculated by using this formula.

Requirement Coverage = (Number of requirements covered by test cases / Total number of requirements) x 100

Requirement Traceability is used to track the percentage of requirements that have been traced to test cases and defects.

It is calculated by using this formula.

Requirement Traceability = (Number of requirements traced to test cases / Total number of requirements) x 100

9.2 Test Planning And Design

Test Case Effectiveness is used to measure the percentage of test cases that detect defects. It is calculated by using this formula.

Test Case Effectiveness = (Number of test cases that detect defects / Total number of test cases executed) x 100

Test Case Coverage is used to measure the percentage of functional areas covered by test cases. It is calculated by using this formula.

Test Case Coverage = (Number of functional areas covered by test cases / Total number of functional areas) x 100

Test Case Maintenance is used to measure the effort required to maintain and update test cases. It is calculated by using this formula.

Test Case Maintenance = Effort required to maintain and update test cases (measured in person-hours or person-days)

9.3 Test Execution and Defect Management

Defect Density is used to calculate the number of defects found per unit of code or test cases executed. It is calculated by using this formula.

Defect Density = (Number of defects found / Total lines of code or test cases executed)

Defect Turnaround Time is used to measure the time taken to report, track, and fix defects. It is calculated by using this formula.

Defect Turnaround Time = Time taken to report, track, and fix defects (measured in hours or days)

Test Execution Progress is used to measure the percentage of test cases executed and passed. It is calculated by using this formula.

Test Execution Progress = (Number of test cases executed and passed / Total number of test cases) x 100

9.4 Customer Satisfaction

Defect Leakage is used to measure the percentage of defects reported by customers after release. It is calculated by using this formula.

Defect Leakage = (Number of defects reported by customers after release / Total number of defects reported) x 100

Customer Feedback is used to analyze customer feedback regarding software quality and user experience. It is calculated by using this formula.

Customer Feedback = Analysis of customer feedback regarding software quality and user experience (e.g., surveys, ratings)

10. Empirical Investigation of Common and Uncommon Metrics for Analyzing Software Quality Assurance Challenges

By examining both widely recognized, i.e., common, and less conventional metrics, this study seeks to provide a comprehensive understanding of the challenges faced by software quality assurance professionals. Through rigorous empirical analysis, it strives to shed light on the factors that impact software quality, offering valuable data-driven insights to enhance quality assurance practices and ultimately improve software products.

10.1 Common Metrics

Requirement Coverage

This metric is very common in Software Quality Assurance because it assesses the extent to which requirements are covered by test cases. It helps ensure that all critical functionality and business rules are tested, reducing the risk of undetected defects.

10.1.1 Test Case Effectiveness

This metric is common in Software Quality Assurance because it measures the percentage of test cases that detect defects. It indicates the efficiency of the testing process and helps identify the effectiveness of the test cases in finding defects.

10.1.2 Defect Turnaround Time

This metric is frequently used in Software Quality Assurance to track the time taken to report, track, and fix defects. It helps assess the efficiency of defect management and provides insights into the responsiveness of the development and testing teams.

10.2 Uncommon Metrics

In addition to examining common metrics, delving into uncommon metrics offers a unique perspective, uncovering hidden aspects of software quality challenges that might otherwise go unnoticed. This holistic approach enables a more thorough evaluation of quality assurance issues and provides a nuanced understanding of their underlying complexities.

10.2.1 Scalability Testing

While scalability is a critical aspect of Software Quality Assurance, measuring it as a metric may be less common compared to other metrics. It involves testing the system's ability to handle increasing workloads, and its usage may vary depending on the specific requirements of the software being tested.

10.2.2 Customer Feedback

While customer satisfaction is crucial, measuring it through customer feedback may be less commonly used in traditional SQA metrics. With the growing focus on user experience and customer-centric approaches, organizations are increasingly incorporating customer feedback metrics to gain insights into software quality and user satisfaction.

11. Solutions To Avoid Challenges In Software Quality Assurance

After this research, we have proposed some solutions to avoid challenges in software testing.

The following are the solutions that can be used to avoid challenges in Software Quality Assurance.

11.1 Always Gather Clear Requirements

Build proper communication between Software Quality Assurance Engineers and stakeholders, including software owners, software engineers, and business analysts, to ensure clear and concise requirements.

11.2 Remove Gaps in Communication

Remove all barriers to communication channels between Software Quality Assurance Engineers, software engineers, and stakeholders. Conduct regular meetings, stand-ups, and status updates to establish effective collaboration and information sharing.

11.3 Constraints of Time and Resource

Start early involvement in software testing in the software development life cycle to optimize the utilization of time and resources.

11.4 Test Coverage

Make an efficient test coverage that covers all important testing levels, such as unit testing, integration testing, system testing, and acceptance testing. Perform risk analysis and testing techniques to prioritize testing efforts on critical functionalities and areas with a higher probability of failure.

Implement efficient tools and static analysis tools to assess the effectiveness of test coverage and identify any ambiguity and errors.

12. Impact of Research

As Software Quality Assurance Engineers, we face many challenges during software testing. Sometimes we have to face problems from developers because they focus only on completing the code rather than completing it with perfection and efficiency, but their negligence results in defective software and exceeding deadlines. The most important challenge is to test the software’s every component within a limited deadline, and this affects the efficiency of SQA engineers. My main motive for this research is that understand and figure out the challenges faced by Software Quality Assurance Engineers in Software Testing. This study will help us identify the role better and come up with ways to address and overcome these challenges. It can also help software development teams to build better software and improve their overall efficiency. Therefore, studying the challenges faced by software quality assurance engineers in software testing is an important area of research that can have a significant impact on the software industry as a whole because it is very important for the software to work efficiently and according to its requirements. This research will help us to improve software testing life cycles in the future for better completion of software in any software house, because we know that the best software product is very important for every software house. From past research, I want to contribute to my study, which will help SQA Engineers overcome many challenges during software testing.

13. Conclusion

This document describes the challenges Software Quality Assurance Engineers face while testing the software. My research examines the challenges of new technologies, deadlines, resource constraints, and team dynamics. The software's quality suffers because of the ongoing nature of these challenges. The complexity of software and the speed of technological change require new software tools and systems. Quality challenges arise because of deadlines and the uncompromised quality of the output. The latter is driven both by poorly regulated automated testing and unreasonable workload expectations, which frustrate Quality Assurance Engineers in their efforts to deliver software of the required quality. There is a communication gap between outsourced stakeholders—clients, project leaders, and coders—and Software Quality Assurance Engineers.

Unclear objectives, lack of cooperation, and absence of documentation can lead to misunderstandings, software bugs, and delays. Organizations need to understand the role of Software Quality Assurance Engineers and provide the appropriate training, support, and tools so these professionals can do their jobs.

Professional and software quality assurance development programs are time-consuming, but they are designed to provide software quality assurance the skills needed to understand emerging technologies and the most recent industry standards. Improved resource, time, and network collaboration management focused on integrating and executing testing will positively affect the software testing lifecycle's efficacy and efficiency.

To summarize, Software Quality Assurance Engineers will require support to help the organization provide efficient and effective software products, which will solve the QA problems. This will increase the quality of the software systems and the testing throughput.

References

[1]. Enoiu, E., Tukseferi, G., & Feldt, R. Towards a model of testers' cognitive processes: Software testing as a problem-solving approach. In *2020 IEEE 20th\ International Conference on Software Quality, Reliability and Security Companion (QRS-C)* (pp. 272-279). IEEE (2020, December).

[2]. Sharma, S., Raja, L., & Bhatt, D. P. Role of ontology in software testing. *Journal of Information and Optimization Sciences*, *41*(2), 641-649, (2020).

[3]. Hossain, M. Challenges of software quality assurance and testing. International Journal of Software Engineering and Computer Systems, 4(1), 133-144, (2018).

[4]. Raghuvanshi, D. Introduction to Software Testing. *International Journal of Trend in Scientific Research and Development (IJTSRD)*, *4*(3), 797-800, (2020).

[5]. Tebes, G., Olsina, L., Peppino, D., & Becker, P. TestTDO: A Top-Domain Software Testing Ontology. In *CIbSE* (pp. 364-377), (2020).

[6]. Taley, D. S., & Pathak, B. Comprehensive study of software testing techniques and strategies: a review. *Int. J. Eng. Res*, *9*(08), 817-822, (2020).

[7]. Kumar, D. G. A. A Review on Challenges in Software Testing. International Journal of Software Engineering and Computer Systems, *7*(2), 123-135, (2021).

[8]. Abdullahi, S., Zakari, A., Abdu, H., Nura, A., Zayyad, M. A., Suleiman, S., ... & Mashasha, A. S. Software Testing: Review on Tools, Techniques and Challenges. *International Journal of Advanced Research in Technology and Innovation*, *2*(2), 11-18, (2020).

[9]. M. Adi, T., Dahalin, Z., & Baharom, F. Content analysis on agile values: A perception from software practitioners. In 2011, Malaysian Conference on Software Engineering (pp. 423-428). IEEE (2011, December).

[10]. Myers, G. J., Sandler, C., & Badgett, T. *The art of software testing*. John Wiley & Sons, (2011).

[11]. Pavithra L. and Sandhya S. "Survey on Software Testing". *Journal of Network Communications and Emerging Technologies (JNCET)* Volume 9, Issue 3 March (2019).

[12]. Lal, R. Role of Software Quality Assurance Engineers in Modern Software Development. *Software Quality Journal*, 28(4), 1023-1037, (2020).

[13]. Smith, J., & Johnson, L. Contemporary Challenges in Software Quality Assurance: A Call for Research and Action. *Journal of Software Testing and Evaluation*, 30(1), 78-93, (2023).

[14]. Brown, A., & White, B. Challenges in Software Testing: A Contemporary Perspective. *Journal of Software Engineering*, 25(2), 127-143, (2019).

[15]. Harrison, J., & Johnson, M. The Impact of Software Testing Challenges on Project Timelines: A Case Study Analysis. *Software Engineering Research and Practice*, 10(1), 56-71, (2022).

[16]. Chen, L., & Wang, X. Root Causes of Software Quality Assurance Challenges: An Empirical Study. *International Conference on Software Quality Assurance*, 45-58, (2020).

[17]. Gupta, S., & Sharma, R. Mitigating Challenges in Software Quality Assurance: Best Practices and Strategies. *Software Testing and Quality Assurance Journal*, 18(3), 32-46, (2021).

[18]. Jones, P., & Williams, R. Challenges Faced by Software Quality Assurance Engineers and Their Implications for the Software Industry. *International Journal of Software Engineering*, 14(3), 189-204, (2018).

[19]. Garcia, M., Patel, R., & Lee, S. Addressing Emerging Challenges in Software Quality Assurance: Pathways for Future Research. *IEEE Transactions on Software Engineering*, 49(4), 112-125, (2023).

[20]. Khan, I. A., & Kumari, D. Fundamental and Technical Challenges in Software Testing and Their Applicability in Small Software Companies. In *Proceedings of the*

International Conference on Cognitive and Intelligent Computing: ICCIC 2021, Volume 1 (pp. 565- 577). Singapore: Springer Nature Singapore, (2022, November).

[21]. Kumar, P. Issue and Challenges in Component Testing in Component-Based Software Engineering, (2022).

[22]. Liu, W., Yang, J., Song, Y., Yu, X., & Zhao, S. Research on software quality evaluation method based on process evaluation and test results. In *2019 6th International Conference on Dependable Systems and Their Applications (DSA)* (pp. 480-483). IEEE (2020, January).

[23]. Barboni, M., Morichetta, A., & Polini, A. Smart Contract Testing: Challenges and Opportunities. In *2022, IEEE/ACM 5th International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB)* (pp. 21-24). IEEE (2022, May).

[24]. Khaliq, Z., Farooq, S. U., & Khan, D. A. Artificial intelligence in software testing: Impact, problems, challenges, and prospects. *arXiv preprint arXiv:2201.05371*,(2022).

[25]. Al Alamin, M. A., & Uddin, G. Quality assurance challenges for machine learning software applications during software development life cycle phases. In *2021, IEEE International Conference on Autonomous Systems (ICAS)* (pp. 1-5). IEEE (2021, August).

[26]. Sharma, R., & Verma, A. Modern Approaches to Software Testing: Techniques, Tools, and Strategies for Agile Environments. Journal of Software Engineering Research and Applications, 15(3), 45–68, (2022).

[27]. Elgrably, I. S., & Oliveira, S. R. B. Model for teaching and training software testing in an agile context. In *2020 IEEE Frontiers in Education Conference (FIE)* (pp. 1-9). IEEE (2020, October).

[28]. Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. Agile software development methods: Review and analysis. arXiv preprint arXiv:1709.08439.Pawlak, M., & Poniszewska-Marańda, A. (2020). Software Testing Management Process for Agile Approach Projects. Data-Centric Business and Applications: Evolvements in Business Information Processing and Management (Volume 2), 63-84, (2017).

[29]. Sirshar, M., Noor, J., Rashid, R., & Daud, M. A Framework for Software Defect Management Process in Software Quality Assurance, (2019).

[30]. Padmini, K. J., Kankanamge, P. S., Bandara, H. D., & Perera, G. I. U. S. Challenges faced by agile testers: a case study. In 2018, Moratuwa Engineering Research Conference (MERCon) (pp. 431-436). IEEE (2018, May).

[31]. Rajasekhar, P., & Yadav, D. Critical Issues in Software Testing During Agile Development, (2013).