EasyChair Preprint

№ 8835

Key Issues in Software Testing in Quality Assurance and Recommendations for Improvement

Maria Afzal, Sidra Yousaf and Saleem Zubair Ahmed

September 15, 2022

"Key Issues in Software Testing in Quality Assurance and Recommendations for Improvement"

Maria Afzal (MSSE-F21-010), Sidra Yousaf (MSSE-F21-015)

msse-f21-010@superior.edu.pk, msse-f21-015@superior.edu.pk

Saleem Zubair Ahmed

Saleem.zubair@superior.edu.pk

Superior University of Lahore, Pakistan.

Abstract:

Quality Assurance ensures that the project is executed according to the pre-agreed requirements, standards, and functions. The aim of this type of standardization research is to better understand the complex process of software testing. There are some flaws in existing software practices, such as testing procedures, user attitudes, and organizational culture. All these issues together, such as shortcut testing, shorter test times, insufficient documentation, and more. In this article, we describe the answer to the great difficulty of product testing in quality management. In this article, we describe solutions for testing problems in quality management. In this study, we propose an approach to tackle the above problems.

Keywords: quality assurance, improvement proposals, planning, documentation.

1. Introduction:

Today, software quality has received a lot of attention and there is a lot of emphasis on producing high-quality software products. Software engineering aims to reduce development costs while improving the quality of software products. [4] Making a working software program is now a daunting task. In order to build a good software product, some measures of software quality characteristics must be defined. When dealing with quality assurance (SQA), keep in mind new strategies, tools, processes, and techniques related to the software development process. Most software is the result of years of collaboration between different developers and designers. No one understands the product of the result. If quality assurance doesn't work, the project will fail. [1] Continued complex growth, customer demands, and increasing market pressures, to achieve higher quality require careful selection of a combination of validation and verification methods to produce software assets on time, on budget, and with the right quality. QA methods mainly focus on the last stages of development, such as the performance phase and related evaluation tasks. [2] This document focuses on two areas of the software process, software correction management, and software quality assurance,

and provides a set of basic tools to assist in the implementation of each specific practice. The "Software Development" process is used to describe, monitor, and evaluate the quality of software products during development. The authors describe the application results and show how the proposed customization tools can help set up and set up the installation process, set achievable goals, and evaluate the results more accurately. Software quality assurance has many problems, first of all how to determine the quality of software. There should be a full understanding of what good software is, but the final specification will often change depending on the context in which the system is used. SQA has several components, ranging from components that occur during the software lifecycle phases to components that occur during most phases. SQA is a field critical to the overall success of a project and requires a variety of skills. New domains of information, such as software security and reliability, are added to a set of essential skills. To be effective, SQA must be an independent body. This article describes quality assurance (QA) strategies, including software testing and assessment, which are important to reduce the negative impact of software errors.

2. Literature review:

Current publications offer advice, strategies, and tools for improving software processes. However, due to our lack of a clear and comprehensive understanding of what is involved in software testing, it is unclear how these concepts, strategies, and tools should be used in real-world testing. Software testing focuses more on standards and procedures than on theory. Various factors are mentioned in the literature that influences the assessment method. These features include, for example, integration testing during development, complex administrative testing, communication and interaction between development and testing, and the use and testing of software components. Cost is also a factor in the decision. Test procedures are associated with test methods. For example, risk-based assessment methods determine the scope of testing, which is especially important. The goal of risk-based assessment is actually to narrow the scope of the test and put more effort into critical tasks. Collaboration and communication are essential for the interaction and communication between the evaluation and development processes.

3. Hypothesis:

Our documentation ideas address key issues such as keyboard shortcuts, faster test times, collaboration and scheduling, lack of user engagement, insufficient documentation, insufficient staffing, management support, low validity, and insufficient knowledge. It's about providing strategies for improvement. This document focuses on the above-mentioned improvement factors.

4. Suggested Improvements for key Issues:

This paper presents an approach to address critical issues encountered during software QA testing.

4.1 Programming and coordination

Test plans should be considered early in software development. Testing will only take place in the final phase of the project. This

checklist should be used at every planning stage. Collect important documents, such as previous versions of documents, document requirements, and document quality tips. Adjustments will be made on the following points: The provision of personnel and equipment used for software development must be well planned. Assign obligations for document segments. Team leaders must estimate financial costs during software development. Programming is very necessary for the planning phase. Careful planning is required to decide which prototype to use and when. Document reviews are also needed to assess the weaknesses of previous projects. Full developer-customer collaboration is required for a successful project and document approval process. Decide how you will handle future updates and developments. Rewrite the text if necessary.

Coordination: The inspection and design teams must work together to protect the project from damage. To provide complete customer satisfaction, you need to establish customer communication between the project and the test team.

4.2 Direct Access Test

Many software program managers and software companies find it a difficult process to test. Software testing is a smart and complex task that requires diligent and dedicated software development personnel. To avoid direct access to the test, do the following: Learn about requirements, construction design, interior design details, and other required documentation. You also need to find the people involved in the project and their responsibilities, the reporting requirements, the required criteria, and the planning requirements that define steps such as evacuation and switchover procedures. Possible points for improvement are: The test team must be fully involved in the testing process. Each member of the testing team must follow the testing guidelines established by the software company. Better planning and coordination between test and development teams are needed. The testing team should consider the feedback and seek continuous improvement. The evaluation team should identify key risk factors for the application, and prioritize and determine the scope and limits of the test. Determine test methods and methods: unit, integration, operation, system, load, usability testing, etc. You should also determine the requirements for the test site, such as computer hardware and software. Find requirements for test materials such as communications and other recording/playback tools, coverage analysts, test track, and issue/bug tracking. You must determine the input data requirements for the test that will be used during testing. Identify jobs, job responsibilities, and staffing needs. The testing process requires establishing a schedule, timeline, and key steps. During the test, refer to the documentation of the test program. Before you start testing, you need to create a test case. Set up test sites and test kits, obtain necessary user manuals/references, documents/repair, manuals/installation, guidelines, establish test follow-up procedures, establish logging and archiving procedures, and configure or retrieve test data. The tester has to find the software developed by the developer and install the software to check for errors. After installing the software, run the test and report the result. Track and review issues/bugs and fixes as needed. Maintain and update cycle test programs, test cases, test sites, and test materials.

4.3 Test time reduction

In that case, you need to follow the steps below. Software developers must stick to the schedule to have enough time to test. The time required for each stage of development should always be adhered to. In practice, tests are often underestimated. Designing and coding often take longer than planning or estimating, so effective management is needed to avoid wasted testing time.

4.4 Management support

Excellent principles can only be achieved by implementing an efficient quality control structure. Quality is built into software products through technology and management procedures designed and implemented to ensure quality, schedule, and budget compliance. There are several technologies to update the software. This includes the main goals of software engineering. Key technologies include defining requirements, preventing defects, detecting defects, and eliminating defects.

4.5 Lack of user participation

Users are very important in the testing process. The principles of collaborative application design and group support systems can be used for user interaction and are preferred in software development. They create strong interactions that are possible between users and developers. Developers should draw the user's attention to testing and encourage them to participate in testing activities, system testing, and acceptance testing.

4.6 Insufficient knowledge

The testing team should have a full understanding of the performance of the software under test, the users, and the platform it runs on. Otherwise, the test will take a negative approach and you will miss the areas that matter most to you. Without environmental information, important user requirements can be missed.

4.7 Insufficient staff

The test is truly a team effort and all team members must contribute to its success. Choosing the right team members for your test can have a significant impact on the success of your test. Testing requires team members with development and testing experience. Group leaders must have problem-solving and management skills, as well as the ability to lead teams and coordinate with clients.

4.8 Insufficient documentation

These two forms of documentation, system documentation, and user documentation, are important during the software development process. The following criteria should be improved to avoid incorrect documentation: Check the content of the entire document for gaps. Bad writing and confusion often cause major problems. Therefore, this part needs to be improved. Reader problems, questions, and lack of foresight from others. Documents are created for the writer and his environment, not the user and his environment. Nevertheless, the documentation must be suitable for the user. Focusing on improving the wrong technical level. Documentation is highly dependent on format and structural design. It is also important to index the document correctly (the document contains the correct information, but is difficult to find).

4.9 Low validity

Software testing consists of programming, effort, and time. Software should be tested using software validation and validation techniques to avoid testability. After

development, we recommend that you follow validation testing, black box testing, white box testing, unit testing, integration testing, system testing, and acceptance testing. Validation testing requires peer and peer review of software between customers and developers at different stages of development. A formal technical review of the software quality assurance activities should be performed during the validation testing. Developers should develop software with the characteristics of maneuverability, observability, manageability, degradability, simplicity, stability, and understanding.

5. conclusion:

Testing is a method of testing data, analyzing software output, and performing software implementations. We can support all software testing techniques, procedures, tools, and philosophies. Effective evidence is the administrator's job. Members of the testing team should focus on consensus with customers. This task proposes techniques to address key software testing and quality assurance issues. The following are considered: test shortcuts, faster test times, collaboration and planning, lack of user involvement, insufficient documentation, insufficient staffing, administrative support, and insufficient knowledge.

6. Reference

1. Iqbal, N., & Qureshi, M. (2012). Improvement of key problems of software testing in quality assurance. *arXiv preprint arXiv:1202.2506*.

2. Denger, C., & Olsson, T. (2005). Quality assurance in requirements engineering. In *Engineering and managing software requirements* (pp. 163-185). Springer, Berlin, Heidelberg.

3. Zeineddine, R., & Mansour, N. (2003, November). Software quality improvement model for small organizations. In *International Symposium on Computer and Information Sciences* (pp. 1027-1034). Springer, Berlin, Heidelberg.

4. Taipale, O., & Smolander, K. (2006, September). Improving software testing by observing practice. In *Proceedings of the 2006 ACM/IEEE international symposium on Empirical software engineering* (pp. 262-271).

5. de Souza, É. F., de Almeida Falbo, R., & Vijaykumar, N. L. (2015). Knowledge management initiatives in software testing: A mapping study. *Information and Software Technology*, *57*, 378-391.

6. Chatley, R., & Field, T. (2017, May). Lean learning-applying lean techniques to improve software engineering education. In *2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering Education and Training Track (ICSE-SEET)* (pp. 117-126). IEEE.

7. Hossain, M. (2018). Challenges of Software Quality Assurance and Testing. *International Journal of Software Engineering and Computer Systems*, *4*(1), 133-144.

8. Burnstein, I., Suwanassart, T., & Carlson, R. (1996, October). Developing a testing maturity model for software test process evaluation and improvement. In *Proceedings International Test Conference 1996. Test and Design Validity* (pp. 581-589). IEEE.

9. Elhag, A. A., Elshaikh, M. A., Mohamed, R., & Babar, M. I. (2013, August). Problems and future trends of software process improvement in some Sudanese software organizations. In *2013 International Conference on Computing, Electrical and Electronic Engineering (ICCEEE)* (pp. 263-268). IEEE.