



Testing and Automation are Very Important Aspects of Software Development

Girish M Chikkahonnegowda*

Department of Electronics and Communications Engineering, Visvesvaraya University, Belgaum, India

*Corresponding Author: Girish Chikkahonnegowda, Department of Electronics and Communications Engineering, Visvesvaraya University, Belgaum, India; E-mail: chisav.girish@gmail.com

Received date: 23 October, 2023, Manuscript No. JEEET-23-119546;

Editor assigned date: 25 October, 2023, Pre QC No. JEEET-23-119546 (PQ);

Reviewed date: 08 November, 2023, QC No. JEEET-23-119546;

Revised date: 15 November, 2023, Manuscript No. JEEET-23-119546 (R);

Published date: 22 November, 2023, DOI: 10.4172/2325-9838.1000971.

Introduction

Testing ensures that the software meets the desired quality standards, while automation helps streamline testing and post-processing process. It allows for faster and more efficient testing, analyzing logs reducing manual effort [1].

Test execution, reporting, automation

Testing is a crucial part of the software development lifecycle. It involves verifying and validating the functionality, performance, and reliability of a software application. The goal is to identify any defects or issues before the software is released to the users.

There are different types of testing, such as functional testing, performance testing, security testing, and more. Functional testing ensures that the software meets the specified requirements, while performance testing focuses on evaluating the system's responsiveness and scalability. Security testing checks for vulnerabilities and ensures the software is secure.

Effectiveness

When it comes to effective testing and process implementation, there are a few key factors to consider. First, it's important to have a well-defined testing strategy in place. This includes identifying the scope of testing, setting clear objectives, and outlining the specific testing techniques and tools to be used.

Another crucial aspect is test planning and design. This involves creating comprehensive test cases that cover different scenarios and edge cases. It's important to ensure that the test cases are realistic and representative of how users will interact with the software.

Test automation can greatly enhance the efficiency and effectiveness of testing. By automating repetitive and time-consuming tasks, can save valuable time and resources [2]. However, it's important to carefully select the right tests to automate, as not all tests are suitable for automation.

In addition to automation, Continuous Integration, and Continuous Delivery (CI/CD) practices can streamline the testing and deployment process. With CI/CD, changes to the software are regularly integrated,

tested, and deployed, allowing for faster feedback and quicker resolution of issues.

To ensure effective testing, it's crucial to establish a feedback loop between testers, developers, and stakeholders. Regular communication and collaboration help in identifying and resolving issues early on, improving the overall quality of the software.

Lastly, it's important to continuously monitor and evaluate the testing process. This involves tracking metrics, analyzing test results, and making improvements based on the feedback received. By constantly refining the testing process, can ensure that it remains effective and aligned with the goals of your project [3].

Best practices or example of real-life implementation

To streamline test execution, test reports, process and automate test execution. Enable tester with all information and improve overall efficiency and remove ambiguity.

Testers often tend to run into various problems and scenarios during test execution and there are lot of dependencies to understand the problems and possible resolutions. Also, manual execution is hard, time consuming and apparently very unproductive. Each engineer would follow different report format and the lack of ticket opening guidelines lead to absolute disorder and inconsistencies.

Creating SharePoint and OneNote with documents consisting of all necessary information such as installation of tools, calibration, flashing, test guide and report templates etc. Basically, this will enable testers to instantly refer for any information they may need with respect to testing and to understand in case of doubt. Continuous optimization by providing flexibility to review and update the document real time in case of addition of new test cases or update in the process [4].

Creation of test Report templates which basically consists of information such as highlighting the strengths, limitations, and recommendations. This certainly helps the customer/management to take important decisions about the release of the product in question thus adding value to the customer. Also introducing race charts, trend charts, QC reporting, different color coding for results for customer reports.

Implementation of automation for both performance and functional test campaigns for improved productivity, reducing test cycle timeline. Also, helps tester to focus on test results and initial analysis.

Ticket creation and first level of analysis

Retesting field scenarios in lab: When retesting field test cases in the lab, it's important to replicate the real-world conditions as closely as possible. This means recreating the environment, configurations, and scenarios that were encountered during the field-testing phase.

Start by documenting the field test cases and any issues or observations that were encountered. This will serve as a reference point during the lab retesting process. Then, set up the lab environment to mimic the field conditions as closely as possible. This may involve configuring hardware, software, network settings, and any other relevant parameters.

Next, execute the test cases in the lab environment, following the same steps and inputs as in the field tests. Pay close attention to any

issues or discrepancies that arise during the retesting process. If any issues are identified, investigate them thoroughly to understand the root cause and potential impact.

It's also a good practice to involve the same team members who conducted the field tests, as they have firsthand knowledge and experience with the specific scenarios. Their insights can be invaluable in identifying any differences or variations between the field and lab environments.

Once the retesting is complete, compare the results with the field test findings. If any discrepancies or variations are identified, analyze them to determine the reasons behind the differences. This analysis can help in refining the test cases, improving the lab environment, or identifying any limitations in the lab setup.

Goal of retesting field test cases in the lab is to ensure that the software or system performs consistently across different environments. By replicating the real-world conditions and carefully analyzing any differences, you can gain confidence in the reliability and stability of your product [5].

Best practices or example of real-life implementation

Engineer must travel back and forth to different Field sites to verify issues or collect debug traces to support bug fixing. This will be leading into lot of travel expenses, delays and as well as engineers will get exhausted. Thereby demand of new resources/engineers to support the campaign arouses.

Creating new virtual field test setup/simulation system in lab. By using field logs, recreating same scenarios and automating these test cases in lab test system. This new approach shall save lot of cost, resources, and time since we could verify the fix in house and make sure fix is working as expected even before taking it into field.

Automation-Post processing process

Automation in logs analysis can be a game-changer, by automating the process of analyzing logs, we can save a lot of time and effort. Instead of manually going through log files, automation tools can help extract valuable information and identify patterns or anomalies.

With automation, can set up scripts or tools to parse and analyze logs automatically. These tools can search for specific keywords, filter out irrelevant information, and generate reports or alerts based on predefined criteria. This can be especially useful in large-scale systems where manual log analysis would be time-consuming and error prone.

Automation can also help in identifying and troubleshooting issues more efficiently. By analyzing logs in real-time or on a scheduled basis, you can proactively detect errors, performance bottlenecks, or security breaches. This allows us to take prompt action and minimize the impact on system or application.

Furthermore, automation can aid in log aggregation and centralization. Instead of manually collecting logs from different sources, automation tools can gather logs from various systems or applications into a centralized location. This makes it easier to search, correlate, and analyze logs across multiple sources, providing a holistic view of your system's health and performance.

Overall, automation in logs analysis simplifies the process, saves time, and enables to gain valuable insights from logs more efficiently. It's worth considering when dealing with a large volume of logs or wants to streamline log analysis workflow.

Effective testing and process implementation require a combination of planning, automation, collaboration, and continuous improvement [6].

Best practices or example of real-life implementation

Creation of automation process and defined signatures for issues based on previous log analysis, in this process basically tester would upload logs in the shared drive where automation scripts would process the logs and try to match with already previously identified signatures.

At the end of processing, if there is a match found either it would open/duplicate the issue in bug system else it would create new ticket and add the new signature to the database.

Conclusion

In conclusion, the presented discussion underscores the pivotal role of testing and automation in ensuring the quality, efficiency, and reliability of software systems. Testing, an integral component of the software development lifecycle encompasses various types, each addressing specific facets such as functionality, performance, and security. The implementation of a well-defined testing strategy, meticulous test planning and design, and the judicious use of automation contribute to the effectiveness of the testing process. In essence, the amalgamation of strategic planning, collaborative efforts, and automation paves the way for effective testing and process implementation. The continuous monitoring, evaluation, and refinement of testing processes ensure their alignment with project goals and contribute to the ongoing evolution of software development practices.

References

1. Marick B (1998) When should a test be automated. ISQW 11 26:1-20.
2. Sneha K, Malle GM (2017) Research on software testing techniques and software automation testing tools. ICECDS 77-81 IEEE.
3. Gojare S, Joshi R, Gaigaware D (2015) Analysis and design of selenium webdriver automation testing framework. Procedia Comput Sci 50:341-346.
4. Fewster M, Graham D (1999) Software test automation.
5. Winkler D, Hametner R, Östreicher T, Biffel S (2010) A framework for automated testing of automation systems. ETFA 13:1-4.
6. Ramler R, Wolfmaier K (2006) Economic perspectives in test automation: Balancing automated and manual testing with opportunity cost. AST 85-91.