IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Impact Factor: 7.53

Volume 4, Issue 2, January 2024

Manual Testing Vital Role: A Research Perspective

Dasari Rathna Nagabhushan

Institute of Distance and Open Learning, Mumbai, Maharashtra, India

Abstract*: Software testing is an important component of the software development life cycle (SDLC) that verifies and validates that a software program runs as intended. It entails carefully understanding the program to detect flaws, ensuring compliance with requirements, and ensuring that it matches user expectations. Software testing employs a variety of strategies, including manual testing, automated testing, and a combination of both. Manual testing is an important component of software development life cycle (SDLC), in which testers are humans who complete the test cases without the use of automation tool. It is an organized process in which testers manually check and validate the performance of a software application to make sure that it meets the proper requirements. Manual testing is classified into many groups, including functional testing, regression testing, integration testing, and user acceptance testing. Manual testing involves following developed test cases, examining the application's functionality, and identifying faults or errors by interacting with the product as end users would. This procedure helps to verify that the program is reliable, functional, and usable. Manual testing is especially useful for scenarios where automated testing is either impossible or too costly, such as exploratory or ad hoc testing*

Keywords: Test Case, Test Plan, Test Execution, Regression Testing, User Acceptance Testing(UAT), Functional Testing, Non-Functional Testing, Test Closure, Test Script.

I. INTRODUCTION

In essence, manual testing refersto the careful and hands-on examination of a software application by human testers. Byusing real people to interact with the tool, manually verifies its various features, in preference to automated testing, which uses specialized tools to run predefined scripts. With the use of this human-focused method, testers can replicate user interactions and spot possible problems that software tests could miss by using their imagination, instinct, and knowledge of the field.Of ev ery approach used to ensure the quality of software, manual testing is one of the most basic and significant ones. In this respect, we undertake a research project to learn much about the delicate details, effectiveness, and enduring importance of manual testing in the constantly evolving field of software engineering.In summary, this research aims to explain the variety of manual testing, providing knowledge about its role, effectiveness, and importance in modern software development. By researching how people participate in testing, we hope to deliver vital knowledge that will boost overall software product performance and accelerate testing methods in a quickly changing specialized scenario.

Definition

Manual testing is a primary method in software quality assurance in which individuals do a plan-by-plan and hands-on review of a software application. Unlike test automation, which uses tools to execute pre-scripted scenarios, tester tests the needs real-time interaction with the software to check its functionality, usability, and implementation to given requirements. Testing by hand is useful for scenarios when judgment from people is required in test design, and in the subjective evaluation of user interfaces and experiences. Although automated testing methods are quicker then repeated work, manual testing remains a vital and required part of the testing process since it gives flexibility, adaptability, and an individual-focused approach to recognizing possible errors in software applications.

Importance of Manual testing

Manual testing is vital along the software development life cycle. While automated testing has gained significance, manual testing remains an essential part of testing for many different reasons.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, January 2024

1. Detecting Unknown Bugs: Manual testers are able to find hidden issues that automated tools will detect.They examine the program as if they were actual users, finding bugs that automated checks was able to miss.
2. Making Ensure its User Friendly: While automated tests can verify that software is working according to plan, human testers make sure the user interface is pleasant and straightforward. User experience is verified manually through analyzing elements such as buttons, colors, and layouts.
3. Flexibility and Fast Changes: If development proceeds quickly with updates on a regular basis manual testing comes in helpful. Testers may quickly modify their tests, test new features, and confirm that everything continues to function, providing developers with immediate feedback.
4. The Initial Detective Work: In the beginning, while the software has been totally ready, manual testing occurs. Testers can examine plans, write tests, and identify potential problems soon. This helps to the prevention of significant challenges in the future.
5. Handling Restricted Funds Wisely: Manual testing is a low-cost approach for smaller projects or when budgets are limited. It allows complete verification without the need for very costly automated knowledges, making it readily available for more projects.
6. Unplanned and Exploratory Testing: When doing exploratory and unplanned testing, where testers interact with the software in real-time to find unexpected errors, manual testing is vital. This kind of testing improves the quality assurance process generally by allowing unexpected review, adjusting to right at the moment observations, and quick feedback.

Phase of Manual Testing: The process of manual testing is divided into the following phases:

1. Requirement Analysis: Analyze the software project documentation,specifications and a process are used to figure out the needs and pre- requisites.If you want to prepare an alternative strategy, we need to know if any feature is not testable during this phase.
2. TestPlanning: In real-world situations, the testing process begins with test planning. We identify the activities and resources that will help in achieving the testing goals during this phase. We also attempt to identify the metrics as well as the means of gathering and tracking this information all over the planning phase.
3. Test Design: At this point, "HOW" to test is clarified. This phase includes the following duties. Explain the test situation in full. Creating multiple smaller requirements for the test conditions will improve coverage. Determine and collect the test data. Configure and get the testing environment ready. Generate test coverage and requirements traceability metrics.
4. Test Implementation: The primary task at this level of the STLC is to create thorough test cases. Determine which test cases will be included in the regression suite by prioritizing them. prior to accepting the test cases, the review process must be finished to ensure that they are reliable. Moreover, don't forget to approve the test cases before starting the real execution.
5. Test Execution: This stage of the software testing is where actually execution takes place, as the word "now" implies. However, before starting your execution, make sure that the initial requirements is filled. Execute the test cases and record any mistakes or inconsistent results. Completing your traceability metrics simultaneously will allow you to analyze your progress.
6. Defect Report: Depending on your project and stakeholders' preferences, you can send out a daily, weekly, or monthly report. You can send different kinds of reports (DSR- Daily Status Reports, WSR- Weekly Status Reports), but keep in mind that the content of the report varies based on who receives it. If project managers have a testing background, they are interested in the technical aspects of the project; thus, include technical information in your report (number of test cases successful, failed, defects raised, severity 1 defects, etc.).
7. Test Closure: Tasks for the closing activities include the following: monitoring for the completion of the test, whether all of the scenarios tested are executed or consciously decreased, and whether no severity 1 defects are opened. Organizing a lesson learned meeting and producing a lesson learned document (which includes what went well, where the scope of improvement lies, and what may be improved).

Methods of Manual Testing: Manual testing methods include a variety of ways and strategies for systematically evaluating different parts of an application. Here are some important manual testing methods:

Black Box Testing: In this method, the tester or QA analyst manually executes a number of test cases and only analyzes the functionality of an individual module, method, or, on a regular basis, the entire program. The tester will provide the application's input and do a manual test. If the goal of the experiment is achieved, the tester will go to the following set of inputs and report all findings to the team. If any tests executed on manually entered data fail, the user will notify the development team. Black box testing having an techniques are decision table based testing, pairwise testing, cause effect graphing in testing, state transition testing, use case testing.

White Box Testing: This method requires a manual inspection of the system's internal components, such as designs and coding. The development team will check the code in this section by going through it line by line. If inconsistency or challenges occur, he or she will correct the designs or code. The entire process is completed by hand, and the procedure is effective since individuals manually review the code or design. White box testing having an techniques are Data flow testing, control flow testing, Branch testing, etc.

Grey Box Testing: White-box and Black-box testing are integrated in this technique. In this case, the tester has some knowledge of the internal workings of the program. The application's performance and inner structure will be manually reviewed by the tester. In addition to testing the program by manually giving various test cases, the tester will review the code part. In the event that the input fails at any point, the tester will modify the code element. Useful for both integration and system testing.

Types of Manual Testing: There are two types which is functional and non-functional testing.

1. Functional Testing: Functional testing is a sort of testing that aims to determine whether each application feature meets the software requirements. Each function is compared to the associated requirement to determine if its output meets the end user's expectations. There are 4 levels of functional testing are Unit Testing, Integration Testing, System Testing and Acceptance Testing.
2. Non-Functional Testing: Non-functional testing is a type of software testing that checks the product's non-functional aspects such as performance, dependability, and usability. Functional testing determines whether or not a product achieves what it is supposed to do, whereas non-functional testing determines how well it works. Non-Functional types are Performance testing, Load testing, Recovery testing, Reliability testing, Security testing, Stress testing, etc.

Principles of Manual Testing: When going into manual testing techniques, it is critical to focus on the fundamental concepts that govern the process.

Testingdemonstrates the presence of flaws

Software testing attempts to make the application fail. Software testing reduces the likely outcome of flaws. Software testing only discusses the presence of errors, not their absence. Software testing may determine the presence of errors in a program, but it cannot prove that the program is correct. Software can never be totally bug-free, even after extensive testing. While not all faults may be removed, testing can help to reduce their occurrence.

Exhaustive testing is not possible

A significant amount of testing is the process of reviewing the performance of a program in every configuration and inputs, acceptable or invalid. Extensive testing is impossible since the program cannot test each test case. The software can only test a limited number of test cases, assuming that it is accurate and produces the correct output in each test case. If the program tested each test case, it would be unfeasible to spend a further sum, time, and resources.

Early Testing

Proper testing is required to identify errors in the software. The flaw detected at the early stages of SDLC will be very cost-effective. Software testing usually starts in the initial phase, which means at the requirement analysis phase, for the purpose to enhance software execution.

Defect cluster

A small number of modules contain the majority of the project's errors. According to the Pareto Principle of testing for software, 20% of modules cause 80% of software issues.

Paradox of pesticides
It is not possible to discover new bugs by repeatedly running the same test cases. Thus, in order to identify new bugs, it is required to go over the test cases and add or update them.
Testing is context-dependent
The software development setting influences the testing methodology. Different software types require different kinds of testing to be done. For instance, testing an Android application is not the same as testing an e-commerce website.
The absence of errors fallacy
Software that meets user requirements yet has 99 percent bug-free code is nonetheless unusable. Compatibility with all client specifications is significant, as is the program's 99% bug-free execution.
Testing is a Decision-Based Process:
The risks connected to various features or functionalities should determine the order in which testing efforts should be prioritized. Concentrate testing efforts on areas that are essential to the application's success.
Software Correctness Is Not Proved by Testing
Testing can expose problems in software but cannot guarantee total accuracy. It provides assurance about how the program will behave inside the testing parameters.
Manual testing is enhanced by automated testing
The combination of Both automated and manual testing provides additional test coverage.
Testing Is Iterative
The process of testing is iterative and goes all the way through the software development life cycle.
Purpose of Manual Testing: The purpose of manual testing in software development is to assure the quality, functionality, and user happiness of a software program through focused on people evaluation. The key goals are as follows:

1. Recognizing Flaws: Tracking and recognizing errors early in the development process enables you to prevent significant problems and reduce maintenance costs.
2. Usage-Based Validation: Reviewing the software's user interfaces, usability, and overall user experience to ensure it satisfies user expectations.
3. Flexibility in Development: Allowing the fast adjustment to changes in requirements or software features, especially in agile development scenarios.
4. Comprehensive Test Coverage: Including a wide range of test scenarios to ensure thorough validation of various aspects.
5. Subjective Assessment: Assessing subjective characteristics such as attractiveness and user happiness, and addressing elements that automated technologies may overlook.
6. Constant Improvement and Learning: Promoting continuous learning from testing experiences, which leads to a development of testing procedures over time.
7. Risk Reduction: Identifying and managing potential software development risks at all stages of the development process.

Quality Assurance
Quality Assurance (QA) testing is a procedure used by organizations to ensure that their products and services adhere to applicable legislation and regulations. It is a collection of procedures that organizations employ to avoid difficulties and ensure that their consumers are satisfied with the final product.It consists of systematic actions and processes for monitoring, evaluating, and enhancing deliverable quality. To suit the diverse needs of various enterprises, Quality Assurance is divided into four types: internal Quality Assurance, external Quality Assurance, process Quality Assurance, and product Quality Assurance.

Literature Review
Manually tested software is that which has been tested by a human tester using manual procedures. In this scenario, the tester performs the test cases manually, without the use of any automated testing tools. Manual testing is an important part of the software development life cycle since it ensures that the program meets the required quality requirements.

The study discovered that a hybrid strategy enhanced test efficiency and efficacy over manual testing alone. The authors stated that a hybrid method can increase overall quality of testing by combining the benefits of both manual and automated testing. Overall, the literature review suggests that manual testing have advantages and disadvantages, and the optimum technique will depend on the specific requirements of the software project. For many software development teams, a hybrid method that combines the benefits of manual testing may be the best approach. However, manual testing can be costly, time-consuming, and prone to human error. It is limited, however, in that it is unable to detect problems that require judgment from individuals. A number of studies have compared the usefulness and efficiency of manual testing. A hybrid strategy that combines the benefits of manual testinghas gained popularity in recent years.

Manually Tested Tools

Manual testing is when human testers execute test cases without the help of automated testing technologies. However, there are various supporting technologies that can help improve the effectiveness of manual testing operations, manage testing artifacts, and increase collaboration. Here are some common tools for manual testing:

1. Test Link: It is a web-based test management solution that helps with software quality assurance and is one of the easiest to use. It is available via an internet-connected browser. Features of Text Link are this manual software testing tool supports multiple programming languages. Allow testing on multiple platforms and browsers. These tools allow testers to develop, organize, and arrange test cases. They frequently include reporting, traceability, and test case versioning functionality.
2. Bugzilla: Bugzilla is a mature, powerful, and feature-rich defect tracking system. Defect-tracking solutions help development teams successfully track outstanding bugs, problems, issues, enhancements, and other change requests in their products. Features are Bugzilla shows the whole bug change history. Bugzilla keeps track of inter-bug dependencies and displays them graphically. Bugzilla allows users to connect and manage Bug-supporting files. Bugzilla features an integrated, product-based, granular security schema, which makes it more secure.
3. Jira: Jira is one of the best open-source tools for Agile technique tracking and planning. Development teams use Jira to track bugs and projects, run scrums, and visualise workflows with Kanban boards. Jira workflows help with software planning, monitoring, release, and reporting. These features are used to monitor and control mistakes and malfunctions. Assign tasks and prioritize. Work with your teammates. Create reports and track progress with ease.
4. LoadRunner: LoadRunner monitors server and network resources in order to improve performance. In addition, LoadRunner monitors CPU performance, I/O delay, server issues, and network or client delay. When client/server systems are tested, LoadRunner automatically reports their performance. Features are simulating real-world user behaviour and load. Detects bottlenecks and performance concerns. Scalable architecture designed for large-scale testing.Offers extensive reporting and analysis.
5. Apache JMeter: Apache JMeter is an open-source, Java-only program. The software is used to test web applications performance, functionality,and load. It is used to evaluate load, functional behavior, and performance. Features include several kinds of loads, such as database and web. Incredibly extensible and configurable. integrates with different plugins to provide more functionality. Offers thorough performance reports along with graphs.

How to perform Manual Testing

First, the tester examines all software-related documentation to determine which testing regions to use. The tester examines requirement documents to ensure that they cover all of the customer's requirements. The tester creates test cases in accordance with the specification document.All test cases are carried out manually utilizing black box testing and white box testing.If any bugs are discovered, the testing team notifies the development team.The development team fixes errors and distributes software to the testing team for retesting. Manual testing often involves several essential phases to verify that Document Review:Reviewing all software-related documentation in-depth is the first step in the testing process. Documents pertaining to requirements, design specifications, and other pertinent information are

included in this. Requirements Analysis: In order to understand and catalog all of the features, functionalities, and specifications that the customer has requested, the testing team goes over the requirement documents. This study provides as a platform for future testing initiatives. Test Case Development: The testing team develops test cases based on the requirements paper analysis. These test cases explain the specific activities to be taken, the expected results, and the conditions under which the tests should be carried out. Manual Test Execution:Manual testing involves the testing team running the developed test cases. To accomplish this, you must interface with the software, supply data, and ensure that the observed and anticipated results are consistent. This stage may include both white box testing, which looks at internal code and logic, and black box testing, which looks at functionality through the eyes of the user.
Bug Identification and Reporting: During the manual testing phase, the testing team tells the development team of any defects or problems discovered. Typically, this letter includes detailed information on the fault, instructions for duplicating it, and any other relevant information. Bug Fixing: After receiving bug reports, the development team tackles the issues that were discovered. Following the implementation of the fixes, the software is returned to the testing team for further review. Retesting: The testing team retests the program to confirm that all reported faults have been resolved. This requires performing the relevant test cases linked with the addressed issues.Iteration: The testing and development teams communicate constantly with each other throughout the process. Feedback is given, and the cycle may repeat until the software satisfies the necessary quality requirements if new problems are found during retesting.

Challenges in Manual Testing

Manual testing, as we all know, covers a large variety of application characteristics; yet, it is not without flaws. It takes time, requires substantial resource planning, and is susceptible to human error. Here are some of the challenges with manual testing to assist you better understand the process:

1. Gap between dev and test execution: The most significant barrier in ensuring continuous delivery of software is a delay in getting started. Testing an application after development is completed has an impact on the continuous delivery cycle. Inadequate traditional test practices and testing tools contribute to this.It is never enough to underline that testing should take place throughout the early stages of the SDLC, in simultaneously with development. Immediate feedback and participation will improve software quality.
2. Increased Cost, Time and Resources: Manual testing requires the participation of a dedicated team of testers. To achieve the delivery deadline, the manual testing approach may be extremely repetitive, and testers must work quickly. To do this, a huge number of manual testers must participate, exhaustively checking each component of the application on each device on which it will run. Because there are so many test cases that must be manually tested, this method can be time-consuming.Manual software testing costs more as the program's effort and size rise. To achieve the continuous delivery requirements of a major application, a massive number of manual testers and resources are required.
3. Implementing to changes is difficult:Manual testing is not suited for long-term uses. These apps may receive frequent updates and upgrades. It is tedious and time-consuming to regularly examine all of these changes on each device for the same or different release.It would be difficult to continuously deliver software while also validating that each of the numerous device combinations now in use is operational.
4. Errors caused by mistakes: It goes without saying that we will make mistakes. If an issue is overlooked, it can spread to other phases, lowering the application's overall quality. Automatic testing produces more trustworthy and faster findings than manual testing.When testers run the same tests repeatedly, they risk overlooking a few details. During continuous software delivery, there would not be enough time to correct these problems.
5. Lack of real or close to real Test Environments: Another major challenge is dealing with test environments.It is critical to ensure that testers have access to the actual test environment in order to address any potential user problems. Manual Testers must individually verify that all device/OS/browser combinations are compatible and free of any concerns. Given the large range of device orientations and configurations that are currently accessible, this is a difficult and time-consuming task. It is also tough to manually configure these test settings for each tester.
6. Less Test Coverage: Errors occurring on a single platform can only be checked one at a time. Manual testers frequently overlook a few test cases, which might have an impact on the test coverage overall.

7. Less Collaboration: There is a requirement for cross-functional collaborative teams that exchange process updates as testing proceeds. An agile team must ensure that numerous testers from different teams participate at all times. With so many test cases must be manually tested, there may not be enough time for cross-team collaboration. Immediate feedback will allow developers to quickly solve errors and save time. Manual testing in Agile and Continuous Delivery models is almost impossible.

II. CONCLUSION

This study delves into the complicated topic of manual testing, highlighting its importance in ensuring reliability of software and excellence. By researching various testing types, testing techniques, and the collaborative interaction between testing and development teams, a full understanding of the relevance of manual testing has emerged. Manual testing is an important part of the software development life cycle because it allows human testers to immediately assess program functionality, user experiences, and system reactions. Manual testing's human centered method enables the investigation of unexpected conditions, subjective evaluations, and the refined judgment required in specific testing contexts. The study provided information on the effectiveness of hybrid testing approaches, which combine the advantages of manual testing. These hybrid solutions have been found to boost efficiency and performance, highlighting the importance of collaboration between testing and development teams. The frequent feedback loop formed by error identification, resolution, and retesting highlights the mutually beneficial relationship between one of the primary elements of software development. In essence, this research study claims that manual testing should be considered an essential component of software quality assurance. Its usefulness goes beyond the accurate execution of test cases to encompass the sensitive understanding, creativity, and adaptability that human testers provide to the quality assurance process. Manual testing is an important tool in our goal of robust and dependable software development.

REFERENCES

[1]. (PDF) Optimizing Manual Testing Processes In Software Development: A Manual Tester's Perspective (Researchgate.Net)

[2]. (Pdf) The Impact Of Manual And Automatic Testing On Software Testing Efficiency And Effectiveness (Researchgate.Net)

[3]. A Journey From Manual Testing To Automated Test Generation In An Industry Project | IEEE Conference Publication | IEEE Xplore

[4]. (PDF) Software Testing Techniques: A Literature Review (Researchgate.Net)