IJFANS INTERNATIONAL JOURNAL OF FOOD AND NUTRITIONAL SCIENCES

ISSN PRINT 2319 1775 Online 2320 7876

 UGC CARE Listed (Group -I) Journal Volume 11, Iss 4, April 2022

An Analysis of Software Testing Methodologies and Automation Testing

Aaditya Jain, Assistant Professor,
College of Computing Sciences and Information Technology, Teerthanker Mahaveer University,
Moradabad, Uttar Pradesh, India
Email Id- jain.aaditya58@gmail.com

ABSTRACT: ***Software testing is the process that includes running a software application and identifying any problems or problems so that the finished article will be free of defects. Software testing is the fundamental way to determine the quality of any software package. The number of certification techniques and ways to verify the software before it moves into production and off courses to market has multiplied due to the global advance in technology. The testing process has been changed by automation testing. Consequently, the majority of software testing is carried out using algorithms, which reduces both the number of people using the technology and the potential for operator mistakes. Test cases used in automation make it simple to record and maintain numerous situations. As a result, the performance of software testing depends greatly on the software automation testing method. The purpose of this research is to compare unit work to automated testing while learning about numerous software testing kinds, approaches, and devices.***

KEYWORDS: ***Automation Testing, Quality Testing, Software Testing, Software Program, Technologies.***

1. INTRODUCTION

Software testing [1] is characterized as any kind of testing done on something like a software product. Finding software vulnerabilities is the primary goal of testing or software testing. A bug is a flaw or malfunction that affects how a software program or software platform behaves [2]. Software may be tested to determine if it:

- Complies with all requirements specified during the construction process,
- Gives an accurate output for different components,
- Has the ability to be doing the assignment in the given timeframe or an appropriate duration of information.
- Can function in a variety of settings.

Test cases are a compendium of circumstances used only by software testers to determine whether the product being tested is functioning properly or not [3]. The creation of an automatic test case aids in identifying any constraints or gaps in an application, and the mechanization of software platform testing involves a series of procedures, techniques, and actions which can only be followed to execute each test on the program [4]. The outcomes among these runs may be captured and saved. The automated testing model is observed by the actions listed below in Figure 1:

- Integration testing
- Administering or constructing tests
- Tasked with conducting the tests
- Analysis and interpretation of the test.

ISSN PRINT 2319 1775 Online 2320 7876
Research paper UGC CARE Listed (Group -I) Journal Volume 11, Iss 4, April 2022

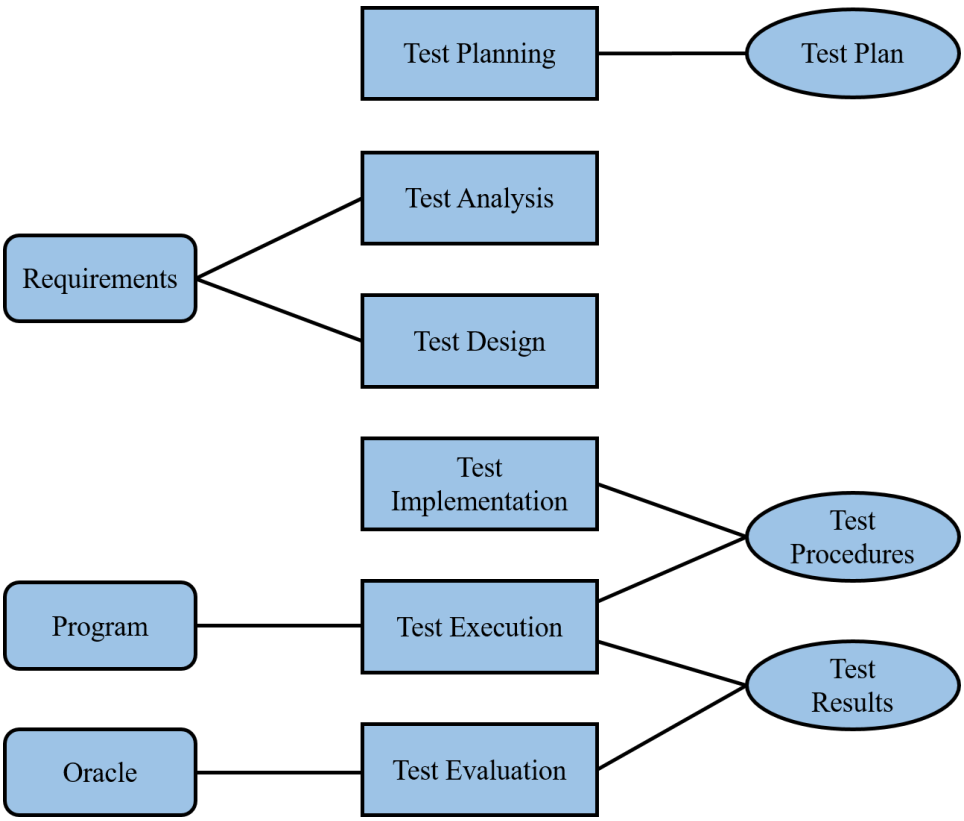

Figure 1: Illustrates That the All Testing processes in Software Testing.

Each activity will do particular tasks that will be employed by another activity in the verification and validation model mentioned above. All bugs will be reported at the culmination of this procedure and may well be preserved as a document [5]. The engineers can identify and take corrective measures thanks to this documentation. The test cases may indeed be divided into online and offline testing as they are created. On the other hand, test automation comes in two flavors. The first determination is based on code and can be described as using pre-existing functionalities, libraries, and classes, but rather modules to test with only a large number of inputs and substantiate and confirm the whether results are correct or not [6]. The second study will mainly focus on a graphical-user-interface (GUI), where mouse clicks are generated and used by the framework to detect changes and evaluate whether a program is performing as designed or not. Manual testing would be laborious and time-consuming with test automation.

1.1. Software Testing Strategies:

The first step to beginning the testing process seems to be to create test cases. To ensure timely and productive testing, the test cases are accomplished by utilizing a variety of detection methods. Black box testing [7], White box testing [8], and Grey box testing [9] are the three main experiment methodologies. Since white box testing analyzes the internal organization of the program in addition to the software's functioning, it is a very productive testing technique. Programming capabilities are necessary to establish the test cases to do white box testing. Clear box and glass box testing are other monikers for white box testing [10]. All testing tiers, particularly unit, integration, and test automation, may use this kind of screening. This sort of testing also known as

ISSN PRINT 2319 1775 Online 2320 7876
Research paper UGC CARE Listed (Group -I) Journal Volume 11, Iss 4, April 2022

security testing meets the requirement to discover if the computer networks secure data and keep performing as expected. Every logical choice is exercised, all loops are verified at each boundary level, and internal data representations are also exercised since this kind of performance testing uses the internal logical layout of the program. As a result, it is capable of testing all the autonomous routes of a module [11]. But since generating accuracy is included in the testing phase, white box testing serves a useful purpose by being a comprehensive testing procedure. A testing method known as "black box testing" basically checks the functionality of the program without going into specifics about its construction. Every stage of testing within the software development life cycle may use this methodology. It primarily carries out the analysis in such a manner that it thoroughly examines every component of the program to ascertain if it matches the user's original needs or not [12]. Examining their functionality at each minimum or maximum, and base case value may detect inappropriate functions. It is the most uncomplicated and widely-used method of testing in use today.

1.1.1. Software Testing Life Cycle (STLC):

The STLC phases, stages, and procedures that a piece of software goes through throughout the testing process are illustrated in Figure 2. Although there is no current standard for software systems undergoing STLC, there are local variations around the globe [13]

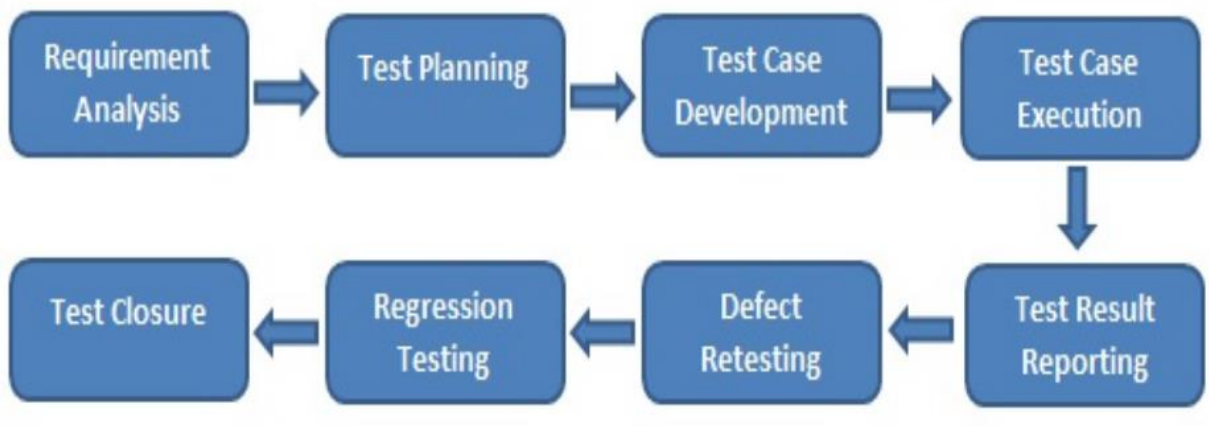

Figure 2: Illustrates the Process of the Software Testing Life Cycle.

The Quality Control team reviews the system specifications during the first phase of the STLC in an attempt to comprehend the fundamental criteria by which the test will be executed. The team members must work with the development team to properly appreciate and resolve any conflicts if they do occur [14]. The second and perhaps most crucial phase of the STLC is test planning, where the whole testing strategy is established? This same creation of the test plan, that phase's final deliverable, is the objective of this stage. Without the Test Plan, the certification process would not be feasible. The Test Plan is a compulsory component that is biased toward the functional testing of the software.

The construction of the test case occurs even during the test designing step, which also spells the end of the test strategic plan. The QA team creates appropriate test situations manually or, in certain instances, automatically. A test case outlines the expected objectives, requirements for

ISSN PRINT 2319 1775 Online 2320 7876
Research paper UGC CARE Listed (Group -I) Journal Volume 11, Iss 4, April 2022

execution, and a collection of test sources or data. The test data set should have been selected such that it yields both the anticipated results and data that is knowingly incorrect because then the test would fail [15]. This is often carried out to ascertain under what scenarios the application goes offline. The test execution stage incorporates running the test cases in conformity with the test plan that was created previously for the implementation process. The test is considered cleared or completed if the functionality is executed flawlessly without any reports of bugs, and then each failed test case will be referenced to the discovered mistake or fault. Such an activity's outcome is a defect or bug report. After the possible solutions have been executed, the collected results are reported. This process also involves problem identification, which is then given to the software development so that it could be rectified.

2. DISCUSSION

Verification is the practice of determining if a certain program meets the criteria that were first provided, or not. It mostly involves the steps of validating and certifying if the created contractor performs the user's expectations. In a conclusion, there is a disagreement between the consequence of this action and what would be anticipated. Finding faults, mistakes, or non-conformance in the program or system that has already been produced is known as software development testing. Therefore, this study gives the customers precise information concerning the product's quality. Software testing seems to be another activity that may be categorized as risk-based. The essential to a successful testing process is for application developers to be possible to decrease a huge number of tests into a manageable test set and decide which risks are crucial to test and which ones are not. The underpinning quality, strategy, and capacities of automation testing tools may evolve depending on the weather or situations they are used for. Therefore, the choosing of automated testing techniques is based on the software expense and the demand for assessment effectiveness for a particular software type. Because we have a diverse array of testing tool alternatives at our disposal, optimization of such automated tools and testing strength is needed. As a result, we discuss certain techniques for using application testing tools in this paper and even the circumstances in which we use them. Comprehensive test automation and management suite that showcases the usage of automated tools have been designed with commercial business several projects and programs in mind. We give the finest ways when selecting our top complex algorithms since testing professionals often fight to discover the best way to test business software and automatically evaluate all project's goals for clients.

3. CONCLUSION

Since the eventual delivery of the product is dependent on testing, it is the most important phase of the development lifecycle. Since it is unskilled labor and time-consuming technique, technological improvements and cutting-edge approaches are important. This makes the integration of automated testing and other test measures both before and throughout much of the testing process. It may strengthen the testing procedures that are being used, both in terms of time efficiency and indeed the creation of a final product that not only exceeds the standards but also offers you the highest operating efficiency. The bedrock upon which software development and testing are conducted is still of extreme significance and is fast changing. However, something as vital as checking sometimes occurs rather late in the software development life cycle. For general comprehension and early review, there should be as much contact as possible between specification authors and testers. This might contribute to solving ambiguity issues and, as a

ISSN PRINT 2319 1775 Online 2320 7876
Research paper UGC CARE Listed (Group -I) Journal Volume 11, Iss 4, April 2022

consequence, reduce the price of future software repair. The penetration tester should provide developers with a particular compact test model after being informed of the criteria and standards so they can ensure their main specifications are satisfied when handling the project for certified testing. The use of computer simulation may greatly aid testers in recreating the environment where the solution will operate so that appropriate exception testing and methods of managing exceptions can be devised. By incorporating simulating into the testing process, it is possible to test the product in a testing environment that is identical to that for which it was designed. Therefore, future work about the testing process will be much more reliant on technology, using simulation and automated testing model-based methodology, speeding up the testing cycle time while also offering the best bug mitigation and effective quality guarantee.

REFERENCES

[1] J. E. Stavesand, S. Reglitz, and A. Himmler, "Optimizing the Benefit of Virtual Testing with a Process-Oriented Approach," *SAE Int. J. Aerosp.*, 2017, doi: 10.4271/2017-01-2114.

[2] Z. A. Pérez, J. A. Schiavon, C. de H. C. Tsuha, D. Dias, and L. Thorel, "Numerical and experimental study on influence of installation effects on behaviour of helical anchors in very dense sand," *Can. Geotech. J.*, 2018, doi: 10.1139/cgj-2017-0137.

[3] Zam-zam fauziah, "pengembangan media pembelajaran berbasis booklet pada mata pelajaran biologi untuk siswa kelas XI MIA madrasah aliyah alauddin pao-pao dan Man 1 makasar," *PLoS Negl. Trop. Dis.*, 2017.

[4] I. ABDULAI, "Assessing The Knowledge, Attitude And Perception Of Hepatitis B Viral Infection Among Young Adults In Sagnarigu District Of The Northern Region," *PLoS Negl. Trop. Dis.*, 2017.

[5] D. A. Wikamorys and T. N. Rochmach, "Application Of The Theory Of Planned Behavior In Generating Patients Intention To Undergo Cataract Surgery," *PLoS Negl. Trop. Dis.*, 2017.

[6] A. Darumeutia, "Pengaruh Gaya Kepemimpinan DanLingkungan Kerja Terhadap Kepuasan Kerja Pegawai PT PLN (persero) Unit Induk Pembangunan Jawa BagianTengah II Skripsi," *PLoS Negl. Trop. Dis.*, 2017.

[7] M. Kumar, A. Professor, S. Kumar Singh, R. K. Dwivedi, and A. Professor, "A Comparative Study of Black Box Testing and White Box Testing Techniques," *Int. J. Adv. Res. Comput. Sci. Manag. Stud.*, 2015.

[8] A. Verma, A. Khatana, and S. Chaudhary, "A Comparative Study of Black Box Testing and White Box Testing," *Int. J. Comput. Sci. Eng.*, 2017, doi: 10.26438/ijcse/v5i12.301304.

[9] R. Jampani, N. Talasu, R. Manjula, and A. B. B. Testing, "Survey of Software Testing Techniques," *Int. J. Res. Appl. Sci. Eng. Technol.*, 2016.

[10] S. Roohullah Jan, S. Tauhid Ullah Shah, Z. Ullah Johar, Y. Shah, and F. Khan, "An Innovative Approach to Investigate Various Software Testing Techniques and Strategies," *Int. J. Sci. Res. Sci. Eng. Technol.*, 2016.

[11] M. Babaeian, V. Ghasemiyan, and R. Nourmandi-pour, "Comparison of software testing review Black Box and White Box and Gray Box," *Int. J. Math. Comput. Sci.*, 2015.

[12] M. Rouse, "Gray Box Testing," *Software Testing fundamentals*, 2010.

[13] M. A. Jamil, M. Arif, N. S. A. Abubakar, and A. Ahmad, "Software testing techniques: A literature review," 2017. doi: 10.1109/ICT4M.2016.40.

[14] I. Akhtar Khan, "Quality Assurance and Integration Testing Aspects in Web Based Applications," *Int. J. Comput. Sci. Eng. Appl.*, 2012, doi: 10.5121/ijcsea.2012.2310.

[15] D. E. Soto Duran, A. X. Reyes Gamboa, and J. Jiménez Builes, "Application of knowledge management to the software testing process," *Espacios*, 2017.