International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6
Article Received: *20 March 2023* ***Revised:*** *17 April 2023* ***Accepted:*** *20 May 2023*

Novel Software Hybrid Testing Model by Using Trusted Computing Group Technology

[1]Kunthumalla Latha, [2]Dr. Nisarg Gandhewar
[1]Research Scholar, Department of Computer Science & Engineering, Dr.A.P.J. Abdul Kalam University, Indore- Dewas Bypass Road, Indore, M.P, India.
[2]Research Guide, Department of Computer Science & Engineering, Dr.A.P.J. Abdul Kalam University, Indore- Dewas Bypass Road, Indore, M.P, India.
Corresponding Author Email : lathaavvar@gmail.com

ABSTRACT: Software testing is a procedure that includes executing a program or application while checking for any errors or bugs in order to produce software that is free of defects. Only testing (software testing) can determine the quality of any software. Worldwide technological advancements have resulted to a development in the number of verification techniques and methodologies available for testing software before it goes to production and possibly makes its way into the market. Thus, automation testing has impacted the testing procedure. Automation tools are used for software testing, which not only minimizes the number of people using the application but also reduces the possibility of errors even with testers testing. Therefore, to increase the efficiency of testing this hybrid model is described. So, by using this testing errors can be detected accurately. This novel software hybrid testing model by using Trusted Computing Group (TCG) technology shows accurate result while testing the software. Hence, this model shows better results in terms of accuracy, time and precision.

KEYWORDS: Software Testing, Trusted Computing Group (TCG), Errors, Software Program/Application

I. INTRODUCTION

Software testing is the process of testing a software product. Finding software bugs is the primary goal of testing, or software testing. A fault or errors that develops from a software program or application is called a bug. Test cases are a collection of conditions that a tester uses to confirm whether the system being tested functions correctly or not. Creating test cases helpful
in identifying an application's requirements and errors.
A number of procedures, tools, and activities must be completed in order to automate any software application testing so that the software can be tested. They can record and store the results of these runs.

Evaluating whether a specific system satisfies its original list of requirements or not is known as testing. The process mostly consists of validation and verification to determine if the created system satisfies the user-specified requirements. Hence, the result of this activity differs from that which was predicted. Software testing is the process of examining developed systems or software to identify errors, bugs, or missing requirements. Thus, the investigation gives the relevant stakeholders accurate data regarding the product's quality. Another way to think of software testing as a risk-based activity includes. The ability to reduce a large number of tests into a manageable test set and to decide which risks are important testing and which ones are not is an essential ability for software testers to possess during the testing process.

A key component of software quality assurance is software testing. Testing for life-critical software, such as flight control, might be extremely expensive because there's a chance of delays in the schedule, complete cancellation or cost overruns. This highlights the significance of testing.

Testing has steps and levels, and the tester varies from level to level depending on those factors. Software testing contains three fundamental steps: unit testing, integration testing, and system testing. The quality assurance engineer, also referred to as a software tester, or the software developer test each of these phases. The Software Development Lifecycle includes all of the testing processes listed above (SDLC). It is essential to divide software development into a number of modules, each of which should be assigned to a different team or person. Unit testing is the process by which developers test each module or unit once it has been completed to see if it functions as predicted. Integration testing is the second step in the SDLC testing process. Following is the separate development of every software system module, the modules are combined together, and this is frequently followed by build errors. System testing, the final testing phase in the SDLC, involves checking the entire application from every perspective.

Software testing also makes sure that no other module's programming is disrupted or interfered with by integrated units. To test every scenario and combination possible, however, when additional components are added to the programs, therefore testing large or extremely complex

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6
***Article Received:** 20 March 2023 **Revised:** 17 April 2023 **Accepted:** 20 May 2023*

systems can be difficult and time-consuming process. For premium optimization, better software testing procedures are also required. The phases of the testing cycle are as follows: test planning, test results analysis, and testing. The primary purpose of test planning, selecting all of the test activities that will be conducted during the testing process is the first step. The second stage of the testing life cycle is called "test development", during which the test cases that will be used for the procedure are developed.

Performing the test cases is the next step in the testing cycle, referred to as test execution. The relevant bugs are then reported during the test reporting phase. "Test Result Analysis", the last phase in the testing process, is where the software developer or the system examines errors. The client is able to take part in this step as it will help both parties better understand what needs to be fixed, improved, or simply modified, and what should be ignored.

Every activity in the testing process model will do certain activities that are utilized by another activity. By the time this process is over, every bug will be reported and saved as a document. The developers can identify errors and fix them with the use of these docs. Test cases can be classified as either automatic or manual once they are developed. Two types of test automation are possible: Testing based on code involves testing with n numbers of inputs and validating and verifying whether or not the results are right using pre-existing interfaces, libraries, classes, and modules. Graphical User Interface (GUI)-based testing: The framework can generate events based on keystrokes and mouse clicks that are used to detect changes and confirm that the program's functionality works as generated. Without automation, manual testing would need a lot of labor and rigorous work.

II. LITERATURE SURVEY

Y. Zhao, Y. Hu and J. Gong, et.al [11] explains that software quality and testing are standardized internationally and examines the connection between the two processes. Finally, it provides a summary of the issues with software quality and testing requirements as well as recommendations. Demanders, developers, independent evaluation parties, and personnel in charge of quality assurance and control will find this paper useful in understanding and applying relevant software quality and testing standards.

A. Mori, et.al [12] describes a strategy that uses code smell data to direct software testing efforts. According to our hypothesis, code smells can help prioritize test cases and reduce the amount of work required for software testing. Software testing is a quality exercise designed to expose bugs that haven't been found yet. Additionally, this task checks to see if the software implements its features in accordance with its specifications. A significant number of test cases may be produced as software testing activities develop throughout software development, which might compromise efforts made during regression testing.

Y. -c. Wang, W. Liu, P. Chen, C. Zhang and C. -j. Li, [13] Software Failure Modes and Effects Analysis (SFMEA) and Software Fault Tree Analysis (SFTA) reverse synthesis-based Field Programmable Gate Array (FPGA) software security test excitation random generation technology is shown. The FPGA software fault tree should first be established using SFTA. Secondly, the SFMEA is carried out for the significant bottom event, evaluating the possible impact of the fault and completing the fault tree. Next, based on the bottom event, adjust the test constraints. Finally, use the verification language to develop the security random test stimulus. This approach can standardize the testing procedure, finally ensure software quality and security by improving the efficiency of FPGA software security testing.

S. Masuda, Y. Nishi and K. Suzuki, et.al [14] provide a technique for testing analysis for complicated software that makes use of international standards for software architectural definitions. We believe the software-test-architecture is the result of the suggested approach. The software-test-architecture supports in the preparing, development of test cases, and analysis of complicated software. We examine approaches to implement the suggested approach to the case studies, using a selection of current case studies that have testing on large and complicated software.

Jagdish Singh, Monika Sharma, et.al [15] gives a thorough explanation of the different web-based application automated software testing tools. Testing is crucial in software development due to increasing complexity and time consumption. Web-based applications require good quality due to economic relevance. Automated testing tools can reduce manual testing efforts and time. Choosing the best tool depends on factors like ease of installation, learning, and performance cost.

L. Deng, J. Dehlinger and S. Chakraborty, et.al [16] It is difficult to find real-world software quality assurance and testing experiences that are authentic and relevant to a Computer Science or Software Engineering program. In order to provide students with useful, hands-on software testing knowledge and experience, this paper discusses five increasingly complex learning interventions which utilize Free and open source software (FOSS) in a software testing course. These interventions are founded on previous experiences teaching Software Engineering fundamentals using FOSS as an instructional medium. Undergraduate and graduate software testing classes will be used to test and evaluate the suggested methodology.

Amira Ali, Nagwa Badr, et.al [17] presents a Testing-as-s-Service framework for web applications that addresses issues like resource utilization and constant monitoring for ensuring reliability of the system. This includes the

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6
Article Received: *20 March 2023* ***Revised:*** *17 April 2023* ***Accepted:*** *20 May 2023*

development and execution of test cases automatically. The performance testing is a important aspect of software engineering, demonstrating how web applications behave under specific workloads. Cloud computing can be used to overcome traditional testing deficiencies by providing Testing-as-s-Service (TaaS) services. This model automates testing activities on demand, reducing time and cost.

Shreya Purohit, et.al [18] Two testing tools are utilized, which are worked on and implemented on a web application: Web Application Performance Testing and NeoLoad (for load and stress testing). Websites and web applications using web interfaces are able to perform load, stress, and performance testing with WAPT. On the other hand, NEOLOAD is a software solution for load and performance testing that is intended for web applications to observe server activity. Adoption will be helped by the evaluation and comparison of various testing tools performances.

Clegg B. S., Rojas J. M. and Fraser G., et.al [19] provide a mapping between the different puzzle categories found in the game's framework and fundamental developer testing techniques like branch coverage and statement testing. Players incrementally learn and practice testing ideas as they advance through the game's levels. We believe that by making software testing seem like an enjoyable task, learners will not only improve their testing abilities but also, over time, develop into better software engineers. Software developers consequently frequently view testing as a liability, which reduces the quality of the program as an overall. One of the reasons for this is that, in contrast to the creative programming and design activities that characterize training, standard testing approaches are frequently seen as difficult and boring.

Valle-Gómez K. J., Medina-Bulo I., Delgado-Pérez P., and Magallanes-Fernández J., et.al [20] describe the collaboration being undertaken by Navantia and the University of Cadiz. Leading Spanish shipbuilding Navantia creates large-scale industrial software applications. Through automating the software testing stage, this project aims to reduce the overall cost of these projects. Every component of the systems must change more quickly than in earlier times due to this new industry. Under such cases, validation and verification become an important stage in the software project life cycle. The consequences of a software failure could be catastrophic, hence it is essential that every system functions properly in production.

III. NOVEL SOFTWARE HYBRID TESTING MODEL BY USING TRUSTED COMPUTING GROUP TECHNOLOGY

In this section, block diagram of novel software hybrid testing model by using trusted computing group technology.

Although it can also be used to consider overall results or additional topics that can have a response along a rating scale, target assessment is especially useful for evaluating several aspects of an event. To compare the endpoint's results under various test conditions (such as treatments), a statistical test is used. Usually, two therapies are used. The study design is paired (dependent) if the results can be acquired for every patient under every experimental condition.

Manual testing is the process of verifying if a manufactured product is of a high quality and whether it is operating in accordance with the requirements. Dynamic instrumentation allows you to add instrumentation into your running production systems without any restarts and at any location in your application's code, including third-party libraries.

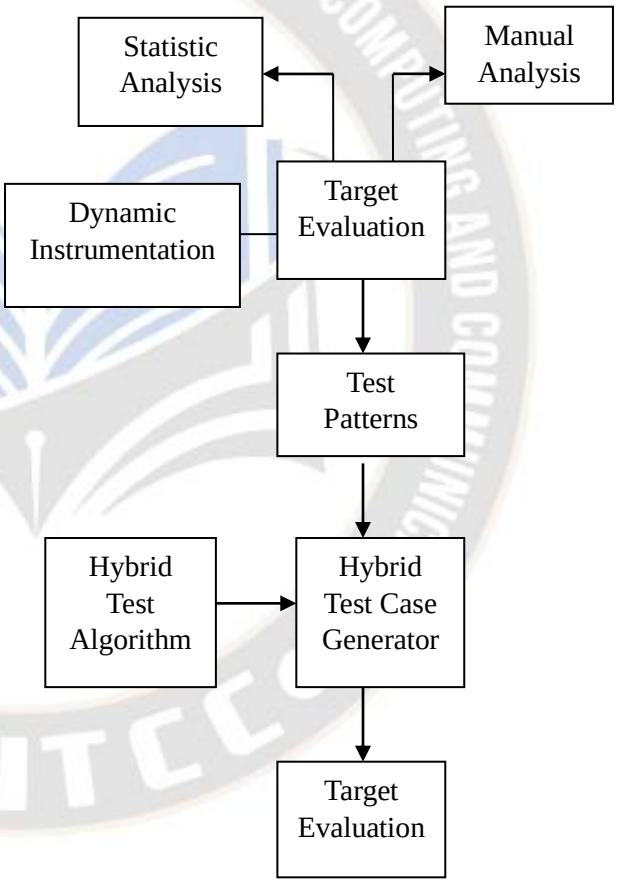

Fig.1: Block Diagram of Novel Software Hybrid Testing Model By Using Trusted Computing Group Technology

An object configuration required to test the interactions between classes that are integrated in accordance with the design specified by the pattern is defined by the test pattern that goes along with a design pattern. A hybrid test is a Quality assurance (QA) strategy that utilizes the advantages of both in-lab testing and crowdtesting to produce an improved software testing campaign. The process of

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6
Article Received: *20 March 2023* ***Revised:*** *17 April 2023* ***Accepted:*** *20 May 2023*

creating test suites for a specific system is called test case creation. The process of creating test suites for a system from a model that describes the system is known as "model-based testing," or MBT. Therefore test evaluation is generated accurate result.

By using different types testing models a hybrid algorithm is described. Hence, by using this hybrid algorithm, this target evaluation generates accurate output.

IV. RESULT ANALYSIS

In this section, performance analysis of block diagram of novel software hybrid testing model by using trusted computing group technology is observed.

Table.1: Performance Analysis

Parameters	Software Testing	Hybrid Software Testing
Accuracy	92.1	99.6
Time	8854	6491
Precision	91.9	97.3

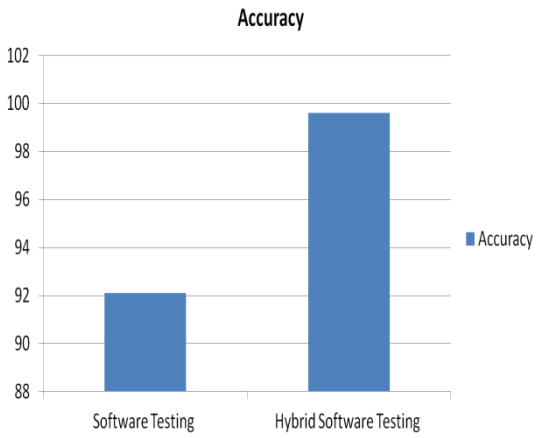

Fig.2: Accuracy Comparison Graph

A comparison graph of accuracy between software testing and hybrid software testing can be observed in Figure 2. Hybrid Software Testing shows higher accuracy.

A time comparison graph between software testing and hybrid software testing may be observed in Figure 3. Hybrid Software Testing takes low time for finding errors.

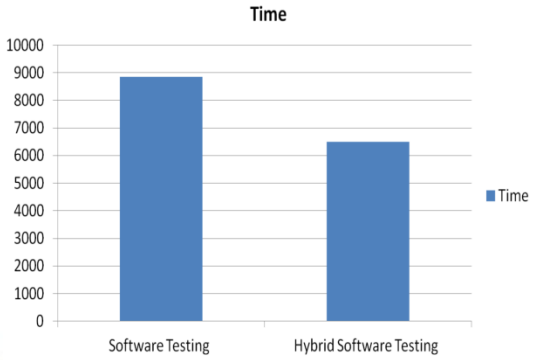

Fig.3: Time Comparison Graph

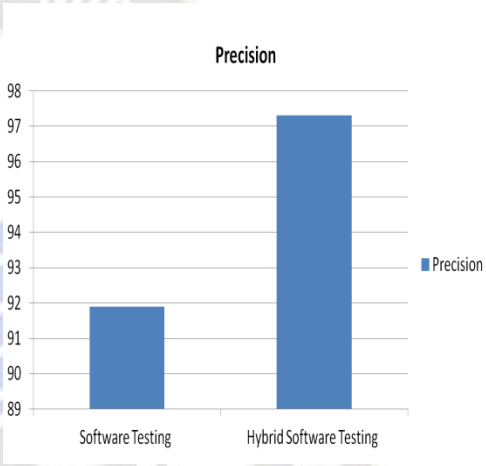

Fig.4: Precision Comparison Graph

A precision comparison graph between software testing and hybrid software testing may be observed in Figure 4. Hybrid Software Testing shows higher precision.

V. CONCLUSION

Therefore, novel software hybrid testing model by using trusted computing group technology is concluded in this section. Only testing (software testing) can determine the quality of any software. Worldwide technological advancements have resulted to a development in the number of verification techniques and methodologies available for testing software before it enters production and possibly makes its approach into the market. This novel software hybrid testing model by using Trusted Computing Group (TCG) technology shows accurate result while testing the software. Hence, this model achieves better results in terms of accuracy, time and precision.

REFERENCES

[1] Cartaxo, E. G., Machado, P. D. L., and Neto, F. G. O. (2017). On the use of a similarity function for test case selection in the context of model-based testing. Softw. Test. Verif. Reliab., 21(2):75–100

[2] L. Yin, S. Zhi-An and Jiang-Ting-Ting, "A Software Reliability Test Suite Generating Approach Based on Hybrid Model for Complex Equipment System," *2017 International Conference on Dependable Systems and Their Applications (DSA)*, Beijing, China, 2017, pp. 144-154, doi: 10.1109/DSA.2017.32.

[3] Antoniol, G., Briand, L., Di Penta, M., and Labiche, Y. (2015). A case study using the round-trip strategy for state-based class testing. In Software Reliability Engineering, 2002. ISSRE 2003. Proceedings. 13th International Symposium on, pages 269–279.

[4] Weissleder, S. and Schlingloff, H. (2014). An evaluation of model-based testing in embedded applications. In Software Testing, Verification and Validation (ICST), 2014 IEEE Seventh International Conference on, pages 223–232

[5] Pooja, M.Tech, UIET, MDU, Rohtak, Haryana, Comparative Analysis of Load Testing Tools Sahi And Selenium, International Journal of ComputerScience and Mobile Computing, Vol.5 Issue.7, July- 2016, pg. 55-60

[6] WANG Ying-qiang, CHEN Sui-yang, WANG Zheng-feng. Design and Implementation of Reusable Software Development Framework Based on .Net[J]. Computer Technology and Development, 2014, (6):122-126.

[7] Singh VB, Chaturvedi KK, Khatri SK, Modeling forecast errors using the complexity ofcode changes ", International Journal of Engineering System Insurance and Management,Vol. 6, No. 1, pp. 44-60, 2015.

[8] S. Prykhodko, N. Prykhodko, L. Makarova, K. Prykhodko, A. Pukhalevych and T. Smykodub, "A Non-Linear Regression Model to Estimate the Number of Defects for Software Testing Phase," *2019 IEEE 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON)*, Lviv, Ukraine, 2019, pp. 965-969, doi: 10.1109/UKRCON.2019.8880027.

[9] H. Okamura, Y. Takekoshi and T. Dohi, "Fine-Grained Software Reliability Estimation Using Software Testing Inputs," *2015 IEEE International Conference on Software Quality, Reliability and Security*, Vancouver, BC, Canada, 2015, pp. 85-92, doi: 10.1109/QRS.2015.22.

[10] G. Joshi and P. Desai, "Building Software Testing Skills in Undergraduate Students Using Spiral Model Approach," *2016 IEEE Eighth International Conference on Technology for Education (T4E)*, Mumbai, India, 2016, pp. 244-245, doi: 10.1109/T4E.2016.061.

[11] Y. Zhao, Y. Hu and J. Gong, "Research on International Standardization of Software Quality and Software Testing," *2021 IEEE/ACIS 20th International Fall Conference on Computer and Information Science (ICIS Fall)*, Xi'an, China, 2021, pp. 56-62, doi: 10.1109/ICISFall51598.2021.9627426.

[12] A. Mori, "Anomaly Analyses to Guide Software Testing Activity," *2020 IEEE 13th International Conference on Software Testing, Validation and Verification (ICST)*, Porto, Portugal, 2020, pp. 427-429, doi: 10.1109/ICST46399.2020.00055.

[13] Y. -c. Wang, W. Liu, P. Chen, C. Zhang and C. -j. Li, "FPGA Software Security Testing Excitation Random Generation Based on SFMEA and SFTA," *2019 6th International Conference on Dependable Systems and Their Applications (DSA)*, Harbin, China, 2020, pp. 465-469, doi: 10.1109/DSA.2019.00074.

[14] S. Masuda, Y. Nishi and K. Suzuki, "Complex Software Testing Analysis using International Standards," *2020 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, Porto, Portugal, 2020, pp. 241-246, doi: 10.1109/ICSTW50294.2020.00049.

[15]Jagdish Singh, Monika Sharma, Panjab University, Chandigarh, India, A Comprehensive Review of Web-based Automation Testing Tools, InternationalJournal of Innovative Research in Computer and Communication Engineering, Vol. 3, Issue 10, October 2015.

[16] L. Deng, J. Dehlinger and S. Chakraborty, "Teaching Software Testing with Free and Open Source Software," *2020 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, Porto, Portugal, 2020, pp. 412-418, doi: 10.1109/ICSTW50294.2020.00074.

[17] Amira Ali, Nagwa Badr, Department of infonnation Systems Ain Shams University Cairo, Egypt, Performance Testing as a Service for Web Applications,2015 iEEE Seventh international Conference on intelligent Computing and information Systems (iCiCiS' 1 5)

[18] Shreya Purohit, JECRCU, Jaipur, Rajasthan, A Sustainable Approach for Analyzing Load Testing Tools, Volume III, Issue VI, IJLTEMAS, June 2014

[19] B. S. Clegg, J. M. Rojas and G. Fraser, "Teaching Software Testing Concepts Using a Mutation Testing Game," *2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering Education and Training Track (ICSE-SEET)*, Buenos Aires, Argentina, 2017, pp. 33-36, doi: 10.1109/ICSE-SEET.2017.1.

[20] K. J. Valle-Gómez, P. Delgado-Pérez, I. Medina-Bulo and J. Magallanes-Fernández, "Software Testing: Cost Reduction in Industry 4.0," *2019 IEEE/ACM 14th International Workshop on Automation of Software Test (AST)*, Montreal, QC, Canada, 2019, pp. 69-70, doi: 10.1109/AST.2019.00018.