



International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijareeie.com

Vol. 6, Issue 1, January 2017

A Review on Software Testing

Jitendra Kumar Dubey

Department of Computer Science and Engineering, Galgotias University, Yamuna Expressway Greater Noida, Uttar Pradesh, India

Email Id: jitendra.dubey@galgotiasuniversity.edu.in

ABSTRACT: A good product is the ultimate necessity of each person. There is a need to switch to testing to produce great software. Software testing in application development operation is very essential. It is one of the important tasks that require a great deal of time and skilled labour. The software uses various testing strategies to locate errors. Software Testing might be a way of finding bugs when running a program and the objective is to get a device with zero faults. It's structured to determine software's ability or functionality. Testing software programs is a critical suggestion that of obtaining software system efficiency. Even though many advances are erased out of structured ways and strategies of identification, there is a need to still have software properly tested before it can be treated to the customer's aspect. It is not possible to build software that doesn't need to be modified in its lifetime. The testing process must also change along with it as the program develops.

KEYWORDS: Errors, Software Testing, Software Tools, Quality.

I.INTRODUCTION

Software testing is an essential stage in the process of software development. Testing relates to the software execution process or the portion of the software with the goal of finding the flaws. Testing can portray errors; it could not prove its lack. Software testing is a quite expensive undertaking and occupies about a third to a half the cost of typical business development[1]. The significance of software testing grows with software increasingly evolving into our everyday lives. Software testing has become one of the vital stages of software development. Software testing is performed to assess the goods or utility standards being tested[2]. Software testing can also provide an unbiased, impartial analysis of the software so that the company can respect and understand the dangers of introducing software. Testing techniques involve performing a program or implementation with the purpose of detecting errors in the software. Software testing is an essential instrument for evaluating the software to evaluate its performance. Testing is defined as an assessment process that either the particular system fulfils or fails to meet its initially defined specifications. It is primarily a process involving the process of verification and authentication that whether the program evolved fulfils the user-defined criteria[3]. Software Testing relates to finding bugs, mistakes or lacking specifications in the program or software created. Therefore, this is an inquiry that gives the investors the intimate knowledge about the item's consistency. Software testing can also be seen as an operation based on risk. The crucial thing the software testers have to comprehend during the testing process is how to minimize a huge series of tests into controllable tests set, and make good choices about the dangers that are crucial for testing or not.

Need For Testing

There are several major reasons for evaluating software. Next, software testing is used to assess the software's consistency and appropriateness. Second, it is used to expose the software bugs. And ultimately people are imperfect which we all make errors, some of those errors are irrelevant but others are risky and costly. There's a need to test more or less everything that we create because something may go incorrect — individuals still make errors.

Failure, Fault and Error

Failure- Failure is a system's dysfunction to perform the necessary functions as per its design. If the fault occurs then it is said that the malfunction happens. The exit from the necessary conduct of the system is said to be a failure. It occurs when a defective strand of code is achieved which leads to an inaccurate state which spreads to the production of the program.

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijareeie.com

Vol. 6, Issue 1, January 2017

Fault- One fault is the result of a mistake. It is an error representation, where representation is the form of expression like descriptive document, Data Flow Diagrams, code, etc. Sometimes, it can be said as a flaw. It is demonstrative of one or more errors.

Error- Individuals can make errors in their behaviour or the goods which arise from their behaviour. It can happen just about anywhere. When evaluating the errors in one's thinking behaviour and produced items, it should return to the testing process. The mistakes committed when coding is alluded to as bugs.

II. DATA FLOW IN TESTING

Testing is an operation that determines and enhances the quality of the software by eliminating mistakes in it. The objective of testing is, therefore, to systematically identify different categories of mistakes in the least amount of time and with the least effort. The Flow of Data in Testing is shown below in Fig. 1 Data Flow in Testing

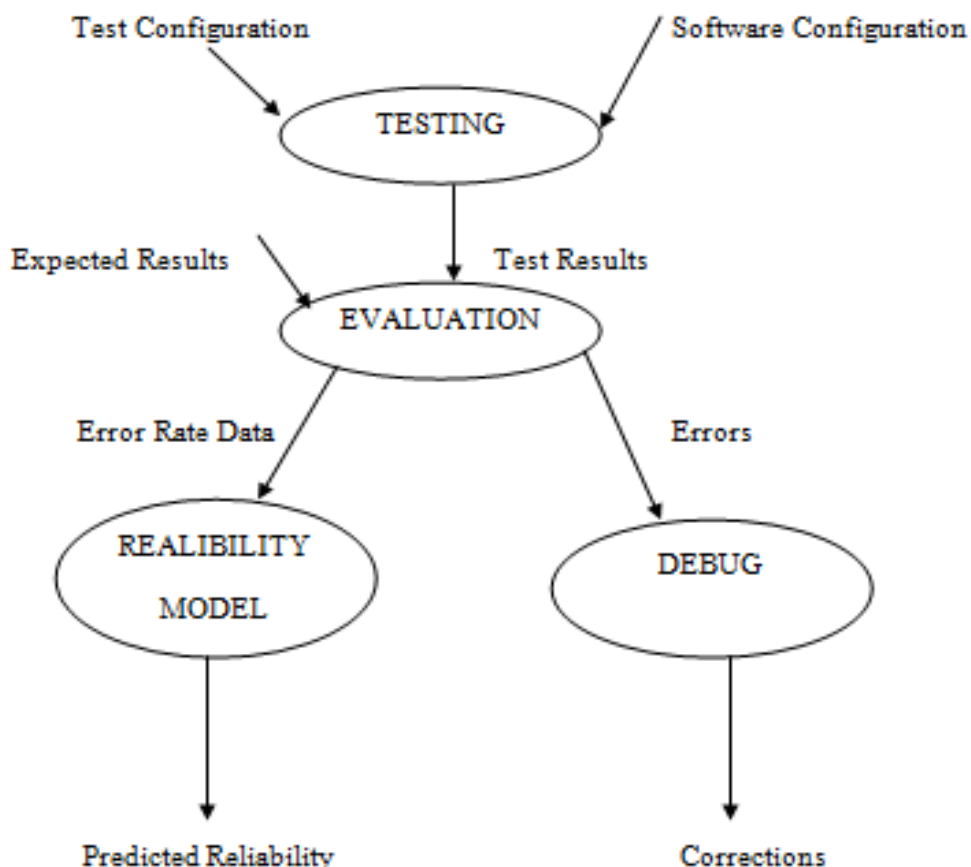


Fig. 1: Data Flow in Testing



International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijareeie.com

Vol. 6, Issue 1, January 2017

III.SOFTWARE TESTING LIFE CYCLE

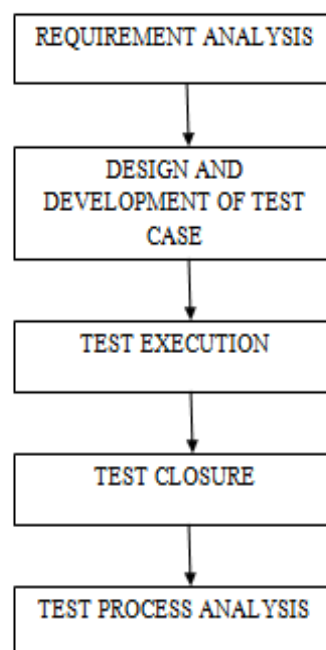


Fig. 2: Software Testing Life Cycle

The Software Testing Life Cycle is shown above in Fig. 2 Software Testing Life Cycle[4].

Requirement Analysis- This is the basic phase where cycle testing starts with an outline of user requirements. The purpose here is to recognize the specifications which are very important for product testing.

Design and Development of Test Case- Which was, what were and what will be the important requirements for the components. The Design specification is examined and tested. Test Specification Feedback is considered.

Test Execution- An examination of the code is performed in this step. Test execution and checking is carried out. And, finally, the simulated outcomes are received according to the efficiency.

Test Closure- A review report of the testing is produced. The project is briefed. Project information is written correspondingly.

Test Process Analysis- Reviews are reviewed with the implementation of new technologies and more functionality to enhance application performance.

IV.TYPES OF TESTING

Tests are often classified by where they are introduced in the phase of software development or by the accuracy level of the examination. Testing is engaged in every level of the software development cycle, but testing is fundamentally different at each stage of software development and has various goals.

Unit Testing- It's finished at the lowest possible level. It measures the software foundation component, which is the smallest component of verifiable software. It is termed module testing or elements testing. This applies to measures that



International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijareeie.com

Vol. 6, Issue 1, January 2017

check the consistency of a particular part of code, typically at the stage of functions[5]. This is normally at the class level in an entity-oriented setting, and the minimum unit checks also contain constructors and destructors.

Integration Testing- It is done if two or more units that have been evaluated are merged into a larger system. The test is sometimes conducted on both the element-to-build implementations and the wider system when its performance assets could not be measured from its elements. Integration testing is some type of software testing that tries to check device implementations against product design[6]. Elements of the software might be integrated one by one time or all components together into an incremental manner.

System Testing- It tends to affirm the end-to-end quality of the entire system. System test is often based on the functional or requirement specification of the system. Non-functional quality attributes, such as reliability, security, and maintainability, are also checked.

Acceptance Testing- Acceptance will be used as a portion of a quality assurance program for performing the functional acceptance of a commodity, facility or system. It is a growing sort of non-operational software testing, which is used mostly for software development and management programs. Such a type of testing depends on the program's operational preparation to be represented.

Alpha Testing- Alpha testing is conducted by possible customers/buyers or an autonomous review group at the developer's side, or real administrative testing. Alpha testing is sometimes used as a type of "internal acceptance testing" for "off-the-shelf software" until the technology goes to the beta testing[7].

Beta Testing- Beta testing occurs following alpha testing and may be taken as a sign of acceptance testing by anonymous users. Software revisions, defined as "beta-versions", are published outside the development team to a smaller audience. The software is distributed to groups of individuals so that testing will guarantee that there are several flaws or glitches in the product. Beta versions are often released to the public, in order to improve the input range to a potential number of possible users.

Regression Testing- It emphasizes on pointing out flaws following a massive code change. Explicitly, this aims to discover software regression or outdated error which has returned. These regressions arise if the functionality of the software that formerly functioned properly ceases to function as expected. Regressions usually take place as an unintended consequence of modifications, when the newly designed aspect of the software intersects with the existing software.

White box Testing- Inner particulars and program structures are created visibly in such a test. Therefore, the identification and resolution of issues is incredibly inexpensive, as a consequence of bugs can usually be detected before they trigger trouble. Such an approach will be outlined as analyzing software using information from its inner structure and programming[8]. This technique is not used widely for testing in massive computer systems and is therefore used in the network service implementations.

Black box Testing- A black box is a device that does not recognize or control the customer's inner data and functions. Such type of testing promotes parameters and performance criteria, and with no inner design information within the software. The primary goal is to verify that the process complies well with the process necessary requirements. Black box testing has less or no data related to the program's internal conceptual model[9]. Therefore, it explores only the fundamental aspect of the system. It guarantees that the information is acknowledged correctly and that outputs are generated correctly.

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijareeie.com

Vol. 6, Issue 1, January 2017

V. TECHNIQUES OF TESTING

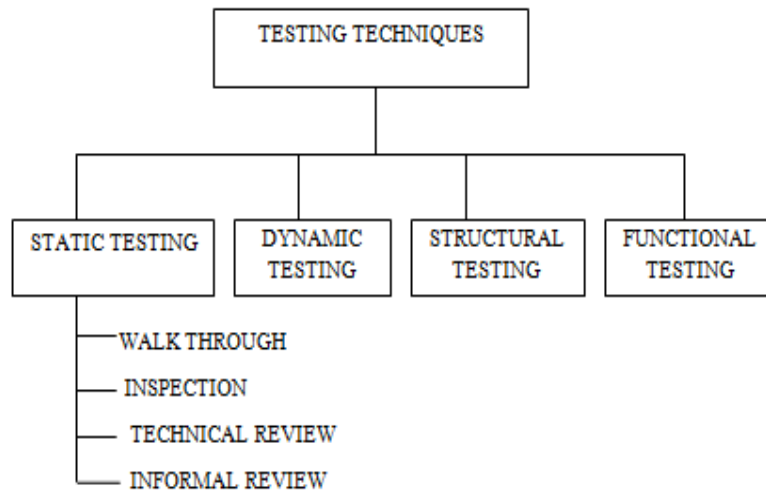


Fig. 3: Testing Techniques

The various testing techniques are shown above in Fig. 3 Testing Techniques

Static Testing-Static software evaluation is the software development evaluation that is carried without actively running applications. The review is conducted on some variant of the source software in several cases and some type of the object code in the certain cases. The various Static Testing Techniques are Inspection, Informal Review, Walkthrough, and Technical Review.

Dynamic Testing- Dynamic testing is also called "dynamic analysis". It is a word used in software testing for describing testing of the program or code's diverse actions. It concerns the evaluation of the program's physical reaction to parameters that are not continuous and vary over time. The program really must be assembled and used in dynamic testing. It includes working with the software, providing initial values and verifying whether the throughput is as anticipated by performing particular test cases that can be performed dynamically or using a computerized procedure. This is as opposed to the static testing.

Functional Testing- Functional testing is a quality control method and a kind of black box testing which models its test cases on the software element specifications being tested. Functions are examined by trying to feed them feedback and investigating the production, and the framework of the inner program is seldom regarded. Functional testing highlights the software entity's exterior actions.

Structural Testing- The software unit is called a white box in structural testing. The choice of test cases is dependent on the software entity being introduced. The aim of choosing such test cases is to trigger different places within the software entity to be implemented, like specific claims, system divisions, or routes. The anticipated outcomes are assessed on the basis of a set of standards for reporting.

VI. TESTING TOOLS

Currently it is possible to get countless Testing Tools kits in the industry. Tool range is fully supported by the company requirements & commercial (Open source / Commercial tools) or open products (Open Distribution Products)[10]. The various types of testing tools are-

- *TET (Test Environment Toolkit)*: TET produces a test driver that accommodates the test group's present and expected long term test requirements. To this end, input from a diverse sample of the society has been used to specify and create the connectivity and devices of Test Environment Toolkit.
- *Test Manager*- The Test Manager is an automatic testing tool for software that is used in daily life testing practices. Such tool is developed using the Java programming code. These test control systems are used to promote, optimize and control regular software design practices.



International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijareeie.com

Vol. 6, Issue 1, January 2017

- *RTH*- RTH is called "Requirements and Testing Hub". This is a software testing tool that we can use as a specification productivity tool as well as providing the error monitoring facilities as well.
- *Ranorex*- This is an easy, detailed and price-effective tool that is used to evaluate automatically. It is a better solution to many other testing tools because, using specific language and prevalent coding strategies, it analyses applications from the user's perspective.
- *Load Tracer*- Load tracer is one of the greatest testing tools for Internet quality created by Trace Innovations Pvt. It is a very user-friendly featured tool used for regression testing and performance testing of web applications.

VII. CONCLUSION

Testing is the most important part of the life cycle of software development because it is something on which the particular product delivery depends. It is time taking, an expensive method thus it needs improved methods and creative methods. Testing is a mechanism in which software quality is measured. It is a work-intensive operation. The testing software uses various kinds of testing. There is the capacity for automation in testing operations but knowledge with testers is quite essential for effective testing. Software testing is part of Quality assurance software. Software system testing is that the commuting part of the development and implementation. Throughout this era of the latest and increased demand for software testing, it is important to continuously illustrate achievement rewards, modern hotspots and suggest different ideas in order to commercialize the software system testing, to promote the rapid growth of software testing. The paper gives a detailed overview on software testing, life cycle of software testing, types of testing, techniques of testing followed by various tools of testing. It is noticed that there are many tools to simplify the research practices for most of the testing activities but there's still scope for developing new and specialized software for testing purposes.

REFERENCES

- [1] P. Ammann and J. Offutt, Introduction to Software Testing, 2016.
- [2] R. Bryce and R. Kuhn, "Software Testing," Computer, vol. 47, no. 2, pp. 21–22, 2014.
- [3] Y. Singh, Software testing, vol. 9781107012967, 2010.
- [4] M. E. Khan and F. Khan, "Importance of Software Testing in Software Development Life Cycle," Int. J. Comput. Sci., vol. 11, no. 2, pp. 120–123, 2014.
- [5] SoftwareTestingFundamentals, "Unit Testing - Software Testing Fundamentals," Software Testing Fundamentals, 2019.
- [6] C. Hobbs, "Integration Testing," in Embedded Software Development for Safety-Critical Systems, 2015.
- [7] A. A. Sawant, P. H. Bari, and P. . Chawan, "Software Testing Techniques and Strategies," J. Eng. Res. Appl., 2012.
- [8] Chinmay Vibhu, "Comparison Study of Black Box and White Box Testing," Int. J. Innov. Res. Technol., vol. 1, no. 12, pp. 2349–6002, 2015.
- [9] softwaretestingfundamentals.com, "Black Box Testing – Software Testing Fundamentals," Softwarefundamentals.Com, 2017. .
- [10] G. O'Regan, "Software Testing Tools," 2019.
- Kavita Arora, Dr. Kavita, Dr. Vishal Jain. (2020). A Study On Attacks In Mobile Ad-Hoc Networks. International Journal of Advanced Science and Technology, 29(8s), 279 - 289. Retrieved from <http://serisc.org/journals/index.php/IJAST/article/view/10502>
- Balamurugan S, Visalakshi P, "Proposing New Strategy for Privacy Preserving Microdata Publishing With Conditional Functional Dependencies", Asian Journal of Information Technology Vol 15, Issue : 12, 2016
- S Balamurugan, K Deepika, RS Venkatesh, R Poornima, Gokul Kruba Shanker, VS Kumar, "SUN Computing: Scalable Ubiquitous Nestle (SUN) Computing for Healing on the IoT", Asian Journal of Research in Social Sciences and Humanities, Volume : 6, Issue : 8, 2016
- RP Shermey, S Balamurugan, "Certain Investigation on Context Aware Knowledge Discovery Strategies for Healthcare Systems", Asian Journal of Research in Social Sciences and Humanities, Volume : 6, Issue : 8, 2016