International Journal of Scientific Research in Computer Science and Engineering
Review Paper Vol-1, Issue-4 ISSN: 2320-7639

A Report on Latest Software Testing Techniques and Tools

Chandraprakash Patidar

Department of Information Technology, Institute of Engineering & Technology DAVV Indore, M.P.-india
chandraprakash_patidar@yahoo.co.in

Available online at www.isroset.org

Received: 07 July 2013 Revised: 18 July 2013 Accepted: 12 August 2013 Published: 30 August 2013

Abstract— Software testing is a process in which software is tested against the requirements of users. Software testing is itself a separate discipline in itself. Software testing is very important because software must be error free before it goes to the user. There is various software techniques available classified on the basis of nature. As technology grows software testing tools are also emerged to perform software testing smoothly. In a competitive market age variety of software tools available, a software tester can use a tool according to their requirements. Generally in software testing the most commonly used terms are validation and verification. Some people consider both terms as same but both term has different meaning and closely related to each other. Software testing is a process of verifying and validating software.

Keywords—Software Testing, Testing Tools, Validation, Verification

I. INTRODUCTION

Software testing, depending upon the testing methods employed can be put into effect at any time in the software development process. An early start to testing impacts the cost incurred, time needed to rework and ultimately results in error free software that is supposed to be delivered to the client. As we know testing is a never ending process and so there is no definite time when one can stop testing and can say that the given software/program is 100% tested and completely fault free. The inherent intendant of software testing is to unmask software wrecks so that the kinks in software may be unveiled and rectified. The testing process cannot clinch that a product functions copiously under all neck of the woods but can only juncture the circumstances or the environments under which it does not function properly. These ways of software testing comprehend the execution but also delve in to miens of the code. In prevalent scenario of software development there are two separate teams, development team and a true blue testing team. Now as number of software packages in the market is increasing, so has the desideratum of the software testers. Until 1980's *"SOFTWARE TESTER"* was the term generally used; but later after seeing the drum re in the IT industry it was seen as a separate profession. Different roles have been established regarding different aspirations in software testing as stated below:

- Manager
- Test lead
- Test analyst
- Test designer
- Tester
- Automation developer
- Test Administrator

II. TYPES OF TESTING

Web Testing: is the name given to software testing that spotlights on web applications. In web testing complete testing of web based software/program/system can help address inflicts and scoop them to the developer before it is revealed to the public. Issues such as the security of web applications, the basic functionality of the site, its accessibility to fully able and handicapped users as well as its readiness for the expected traffic and number of users which is though related to load testing[3]. The *web security testing* tells us whether web based application requirements are met when they are subjected to malicious data. There are two different types of testing which are used to test software as described below:-

Manual Testing: This type of testing includes the testing of the software manually that is performing the testing without using any tool. In this type the tester performs the role of an end user and test the software to get the hang of any unexpected behaviour or bug in the program/software. There are different junctures for manual testing like unit testing, integration testing, system testing and user acceptance testing. Testers use test plan, test cases or test scenarios to test the software and asseverates the plenitude of testing. In this type of testing, the testers explore the software to spot errors in the program/software.

Automation Testing: Automation testing is also known as *"TEST AUTOMATION"*. In this type of testing the tester write scripts and makes use of software to test the software/program. In this type the manual process is done by the automated or the consecrated software. It is also used to re-run the test scenarios those were consummated manually earlier quite swiftly and re iteratively. Automation testing improves the incisiveness, bail out time and money in juxtaposition to manual testing. Also, there are certain snags related to automated testing as it is not possible to automate everything in software, the areas where users can make transactions such as login form or the registration forms or simply the areas where the requirement is automation.

III. SOFTWARE TESTING TOOLS

There are a lot of automated tools already on deck which can be used to write the automation scripts. Some of the tools which can be used to perform automated testing are listed below[4]:-

- HP Quick Test Professional
- Selenium
- IBM Rational Functional Tester
- Silk Test
- Test Complete
- Testing Anywhere
- Win Runner
- Visual Studio Test Pro
- WATIR etc.

IV. DIFFERENT METHODS OF MANUAL TESTING

There are a variety of methods which can be used for testing the software:-

Black Box Testing: It is a testing technique done without having the cognition of the internal workings of the application. The testers identify the system architecture but they do not have equity to access the source code. Basically, black box tester will interact with the system's user interface by providing inputs and examining the outputs without knowing how the inputs are worked upon. The advantages and disadvantages of black box testing are mentioned in Table 1.

White Box testing: It is the complete examination of internal logic and structure of the code. White box testing is also called as *glass testing* or *open box testing*. In order to execute the open box or white box testing on an application, the tester needs to possess insight of the internal working of the source code and find out the module which is incongruous or showing error. The advantages and disadvantages of white box testing are mentioned in Table 2.

Grey Box Testing: In grey box testing the tester has limited apprehension of the internal workings of the application. Getting the knacks of the domain of a system always gives the tester an edge over someone with limited domain knowledge. Unlike black box testing, in the grey box testing the tester only test's the user interface, the tester has access to design documents and the database. Since the ingenious testers have better knowledge; they are able to create better test data and test scenarios while making the test plan.

V. SELECTION OF SOFTWARE TOOLS

Selection of software tools is entirely based upon the project requirements & commercial (proprietary/commercial tools) or free tools (Open source tools). The free tools have some injunctions due to accessibility of some features. The tools are prorated into following different categories[5]:

- Test management tools
- Functional testing tools
- Load testing tools

VI. SOFTWARE TESTING TECHNIQUES

Some of the substantial software testing techniques are discussed below in nutshell.

Equivalence Partitioning: This technique segregates the input of the program onto equivalence classes. Equivalence classes are set of valid as well as invalid states for input conditions. While using this technique, one can get test cases which identify the classes of errors. Boundary Value Analysis: This technique is like equivalence partitioning except that for creating test cases input domains as well as output domains are also used.

Cause Effect Graphing Techniques: When one uses this technique one wants to explicate a policy or procedure specified in a natural language into software's language. In this the cause effect graph is produced and then the graph is changed into a decision table and then finally the rules of the table are modified to create test cases.

Comparison Testing: When the fidelity of software is critical, superfluous software is produced. In this technique, two software teams' out turn independent chronicles of an application and then each version is tested with the same test data and finally same out put is ensured.

Fuzz Testing: It is often called fuzzing or negative testing. This testing feeds random input to the application. The main characteristics of fuzz testing are:

- Input is random.
- If application crashes or hangs, test is failed.
- Fuzz testing can be automated to high degree.

A tool called fuzz tester can be used which indicates causes of founded vulnerabilities.

Model Based Testing: It is a mechanized spawning of efficient test procedures using models of system requirements and specified functionality. The multifarious benefits of model based testing are:

- Early bug detection.
- Can generate endless tests.
- Find crashing and non crashing bugs.
- Automation is cheaper and more reliable.
- One implementation per model then all cases free.
- Testers can address bigger test issues.

Basis Path Testing: This technique is antiquated to evaluate logical intricacy of procedural design. One can employ this measure for describing basic set of execution paths. Flow graphs, cyclomatic complexity, graph matrix and link weight are utilized for obtaining the basis set and presentation flow in the program.

Loop Testing: It is a white box testing technique consummated to validate the loops by proclaiming the loops initialization contingencies. The uninitialized variables in the loop can be clinched by passing through the loop once so as to avoid loop repetition issues and unveiling performance
clogging.

REFERENCES

[1] Kaur, M 2011. Comparative study of automated testing tools. Int.J. Comp. Appl. 24(1): 1-7.

[2] Luo, G and Ural, H. 1995. Approach to constructing software unit testing tools. *Soft. Engg. J.* 10(6): 245-252.

[3] Mesbah, A, Van Deursen, A and Roest, D 2012. Invariant-Based automated testing of modern web applications. *IEEE Trans. Soft. Engg.* 38(1): 35-53.

[4] Rattan, R 2013. Performance evaluation and comparison of software testing tool. Int. J. Inform Comp. Technol. 3(7): 711-716.

[5] Shahid, M 2011. A study on test coverage in software testing. Int. Conf. on Telecommunication Technol. Appl. Vol. 5.

[6] Nirmal Kumar and B G Geetha. 2014. Working with IBM Rational Functional Tester: A Software Testing Tool. Journal of Academia and Industrial Research. 2(12):669-674.