

Software Testing: A Survey of Software Testing Techniques**Prof.(Dr.) A.K.Upadhyay**

Dept. of Computer Science & Engineering, AUMP Gwalior

akupadhyay@gwa.amity.edu

Abstract

A software system development process is cumbersome, complex and is error prone. As software development is phased process, there is every possibility of defects and error. The errors, if not controlled timely may prove to be havoc and very costly in terms of time and money. The bugs need to be identified and fixed as early as possible. Software verification and validation techniques play a very important role software development. With the help of these techniques, errors can be detected early and thus can help in making the software cost effective. Software testing techniques are essential to ensure that the developed software is error free. As the process of testing is time taking, it becomes both time and money consuming. The objective of testers is, therefore, to reduce cost. The cost can be reduced in two ways, by testing only the changed part of software and making use of testing tools.

Keywords: software testing, dynamic testing, regression testing, validation, verification.

Introduction

The psychology of a software tester is to find faults in software. In other , it can be stated as software testing is a process of verifying and validating the developed software meets the software requirement specifications. Effectiveness of software test suites is ability to find errors in software. Efficiency, on the other hand, deals with cost effectiveness of test cases. Reducing the time taken for software testing is essential[1]. Exhaustive testing is not possible and thus software testing is endless. It is a good idea to club software testing measure along with software development process.

Software verification and validation

In order to make the software free of errors, verification and validation techniques are applied.

Verification

Verification is intended to make sure that software meets all the functions stated in software requirement specification document. The techniques such as conducting reviews, meetings, walkthroughs and inspections etc are intended to ensure that the software is error free.

Validation

Software validation typically involves Unit testing, integration testing and the system testing. Validation is done after verification is over[2]. Validation is done at the end of the development process.

Some important aspects of testing

The important thing to know in software testing is when to go about testing techniques, which techniques would be sufficient and obviously when to terminate.

Who should test?

The task of software tester is to build test suites at testing the software against the requirement of software[3,4]. As a broad guideline following members take the responsibility of software testing; Project Manager, Software tester, Software developer, end User.

When should testing begin?

Ideally testing should start along with feasibility study of software. It not only reduces time to fix errors but is cost effective. "Following situations can also be considered as testing:

- Analysis and verifications of requirements during requirement gathering phase.
- Reviewing the design in the design stage.
- Testing performed by a developer on completion of each section of code.
- Indirect testing by the end user after implementation".

When should the Testing be terminated?

The decisive criteria could be any of followings though it could be difficult to specify:

- "All the high priority bugs are fixed.
- The rate at which bugs are found is too small.
- The testing budget is exhausted.
- The project duration is completed.
- The risk in the project is under acceptable limit.

As a matter of fact it can never be said that testing is 100% done still above are a few guidelines[5]. Generally if the stakeholders of the software can manage risk ,it can be stated software testing is done. Risk can deduced by simply: Measuring test coverage and/or Number of test cycles and/or Number of high priority bugs".

Automating the software testing:

The difference between manual testing and automatic testing is discussed in is running test cases manually without tool support is known as manual testing[6]. Using the toolkit and running test cases using automation tool (s) is known as automatic testing.

Manual Testing

Manual testing is the process through which software developers run tests manually, comparing program expectations and actual results to find software defects[7]. These manual tests are nothing more than a tester that uses the program as an end user, and then determine whether the program is working accordingly. Manual testing is well suited for small projects, as well as companies without significant financial resources.

Advantages

1. Software automation tools are expensive. Short-term costs are much lower with the help of guidance testing.
2. Automatic tests do not necessarily act like people. Consequently, some of the the end user may encounter, may be missed in the automatic test. Manual testing for on the other hand, allows you to develop a program that will be used by the end user. The human user can process the program in a special way, which can lead to errors which, most likely, will be missed in the automatic test, but are caught during manual testing[8].
3. Manual tests are flexible than automatic tools. Automated tools are launched according to the set rules[9]. Test cases are configured and programmed in an automated tool, and then tests are carried out. When a change occurs in the project, the entire process may need to be repeated. But with manual testing, this can easily be included in the testing procedure.
4. With manual testing, one can quickly see the results. Automatic tests require more time for and sometimes the whole set of tests must be performed before the results are output[10,11]. Therefore, manual tests allow one to quickly and easily test ideas.

Disadvantages:

1. Some tests are difficult to perform manually, for example, regression of a low-level interface testing is extremely difficult to perform manually[12]. As a result, it is prone to errors and is controlled when it is done manually. Automatic testing, after setting it up, is much better to find errors for this kind of testing.
2. Manual testing can be repetitive and boring - no one wants to continue to fill in the same form over and over again. As a result, many testers are hard at it process and errors are more likely.
3. With automatic testing, if you add data to the program, you can restart all the required tests immediately, because the tests are already set up[13]. This is not so with manual testing. If there are any changes to the software, one must run the tests again.

Automatic testing: It is the process by which automated testing tools run tests that repeat the predefined actions, comparing the expected and actual outcomes. If the program's expectations and results match, your project behaves like should, and you, most likely, will not find mistakes. However, if they do not coincide, there is a problem that it is necessary to solve. You need to look at your code, change it and continue run the tests until the actual and expected results are agreed.

Advantages:

1. Run tests quickly and efficiently: although the initial configuration of automatic test cases can take some time, after you have automated your tests, you are good to go. You can re-use the tests, which is good news for those of you who work with regressions on a constantly changing code. You do not have to constantly fill in the same information or not run certain tests. Everything is done automatically[14].
2. It can be cost-effective in the long term: although automation tools can be costly short-term, but they save your money in the long run. They not only do more than a person can for a certain amount of time they also quickly detect defects. This allows your team to react faster, saving you precious time and money.
3. Filling in the same forms of time over and over again can upset. But automation of testing solves this problem. The process of creating test cases requires coding and thought that keeps your best technical thoughts involved and committed to the process.
4. Everyone can see the results: when one person conducts manual testing, the rest the team cannot see the results of the tests being performed. However, using automated tests; people can enter the testing system and see the results. This allows you to increase the team cooperation and a better final product.
5. Automatic testing is more suitable when the project is large (that is, the number of inputs / configurations are high), there are many system users or when filling out forms.

Disadvantages

1. Automation tools can be expensive to buy. As a result, only use those that will give you full or close to full coverage, as you can find.
2. Although the automation process reduces the time it takes to check everything, automatic testing still requires a lot of time.
3. Tools have limitations. Although automatic tests detect most of the errors in your system, there are limitations. For example, automated tools cannot test visual Such as image color or font size. Changes in them can be detected only manual testing, which means that not all tests can be performed using automatic tools.
4. Methods, levels and types of testing:Methods, levels and types of testing are applicable throughout the life cycle of software testing.

4.1 Methods: There are several approaches / methods for testing software. Although there are many methods of software testing, currently available, in accordance with our the organization

of this document will only include the following methods.

4.1.1 Static testing: Static testing involves checking and usually requests or checks: "You are building is the thing correct? "This is primarily a syntactic code check or a manual code check, document requirements, project documents, etc., to find errors [1].

4.1.2 Dynamic: Dynamic testing involves checking and usually asks for or checks: "You are building the right thing? "The software must be compiled and executed, and input values and the output values are checked with the expected output .

4.1.3 The Black Box: The Black-box test treats the software as a "black box" with inputs and outputs . The tester only knows what the software should do, and not how it does it. Consequently it is also known as a test method based on a specification or testing using I / O methods.

4.1.4 White Box: In tests with a "white box" or "glass box" internal structures or works the program has been tested. It is also known as a structural method, because here testers require knowledge of how the software is implemented. In contrast to testing a black box, The tester concentrates on how it is done in white software .

4.2.2 The component Testing When testing components, individual program modules are tested to find defects and to verify their proper functionality. Component testing is also known as a module and testing the program. Component testing can be done in isolation from the rest of the system depending on the model of the development life cycle chosen for this particular application [12].

4.2.3 Integration testing: When testing integration, individual program modules are combined and tested as a group for make sure that they work together without errors. Integration testing is performed after unit testing and before validation. This is done using a special integration tester or test group [12].

4.2.4 Testing the system The system is tested in a complete integrated system for system evaluation observing the established requirements. Testing the system enters the sphere of black and does not require knowledge of the internal code of the code or logic. System testing examines both functional and non-functional requirements for software [13].

4.2.5 Acceptance testing Once all or most of the defects have been fixed after testing the system, the system will deliver user or customer acceptance testing. It is carried out to determine, these requirements are met before delivery. Acceptance testing is mainly made by the user or client, although other interested parties may also participate [11].

4.2.6 Alpha Testing Alpha testing is modeled or actual operational testing of potential users / customers or an independent test group on the developers site. Alpha testing is often used for ready-made software as a form of internal acceptance testing, before the software passes to beta testing. This happens on the developer's site [14].

4.2.7 Beta Testing Beta testing or field testing, and occurs on the customer's site. Software installed and tested in real working conditions. Software versions known as beta versions are released for a limited audience - for example, a sample of target customers – outside development team [15]. This is very common in game development.

5. Conclusion:

This paper presents an introduction to software testing and its problems. The paper presents two separate articles as solutions for a large number of test cases problem with the use of "Combinatorial Testing" and a huge test time problem with Test Automated. Combinatorial testing can detect hard-to-find software errors effective than manual methods of selecting test cases. While the most basic form combinatorial testing - paired testing - well established, and the introduction of software testing of practitioners continues to grow, the software industry has not yet combinatorial testing. There are advantages and the disadvantages of automated and manual testing. Although many say that common sense should guide you in deciding which one to use. One should sure that resources and the size of the project, as well as the quality of the automated tools, as well as the skills and know-how of the testing team. Testing detected information about software errors. This information will be used by the developers to improve the product. Thus, the purpose of testing products and services is to increase productivity of the company. Testing should be transformed into "engineering productivity" approaches. At the heart of the whole technology lie people with dreams and ideas. Without them, no Internet will not work. Good practice arises when testing software the desire of professionals to bridge the gaps between business, technology, customers and consumers. With more flexible, like hybrid methodologies, testing becomes a design criterion; method to improve performance and product. Testing software with a flexible philosophy is a sign that business and its professionals take care of their work through honest communication with integrity and ethics. Finally, the paper presents methods, levels and types of testing - three main categories the process of software testing. There are several approaches / methods of software and they were discussed in terms of testing methods. The software is tested at each phase of the SDLC and, therefore, there are different levels of testing. Test Type the main focus was on the specific purpose of testing, and in this paper, mainly two main types were discussed: functional and non-functional, and then lists some types of testing under each basic type.

References

- 1) Myers, Glenford J., Corey Sandler, and Tom Badgett. The art of software testing. John Wiley & Sons, 2011.
- 2) Kung, David, and Hong Zhu. "Software verification and validation." Wiley Encyclopedia of Computer Science and Engineering (2008).
- 3) Ammann, Paul, and Jeff Offutt. Introduction to software testing. Cambridge University Press, 2008.
- 4) Qatarun.wordpress.com,. 'Start And Stop Of Testing | Software Testing'. N.p., 2015. Web. 20 Mar. 2015.
- 5) Base36,. 'Automated Vs. Manual Testing: The Pros And Cons Of Each'. N.p., 2013. Web. 21 Mar. 2015.
- 6) Smartbear.com,. 'What Is Agile Testing? | Smartbear Software'. N.p., 2015. Web. 21 Mar. 2015.
- 7) Netsolutionsindia.com,. 'How AGILE Testing Helps In Building Better Software Products'. N.p., 2013. Web. 21 Mar. 2015.
- 8) Collins, Eliane, Arilo Dias-Neto, and V. F. de Lucena. "Strategies for agile software testing automation: An industrial experience." Computer Software and Applications Conference Workshops (COMPSACW), 2012 IEEE 36th Annual. IEEE, 2012.
- 9) Zhang, Sai, et al. "Combined static and dynamic automated test generation." Proceedings of the 2011 International Symposium on Software Testing and Analysis. ACM, 2011.
- 10) Lönnberg, Jan, Ari Korhonen, and Lauri Malmi. "MVT: a system for visual testing of software." Proceedings of the working conference on Advanced visual interfaces. ACM, 2004.
- 11) Mustafa, Khaled M., Rafa E. Al-Qutaish, and Mohammad I. Muhairat. "Classification of software testing tools based on the software testing methods." Computer and Electrical Engineering, 2009. ICCEE'09. Second International Conference on. Vol. 1. IEEE, 2009.



- 12) Sawant, Abhijit A., Pranit H. Bari, and P. M. Chawan. "Software Testing Techniques and Strategies." *International Journal of Engineering Research and Applications (IJERA)* 2.3 (2012): 980-986.
- 13) Arcuri, Andrea, Muhammad Zohaib Iqbal, and Lionel Briand. "Black-box system testing of real-time embedded systems using random and search-based testing." *Testing Software and Systems*. Springer Berlin Heidelberg, 2010. 95-110.
- 14) MacCormack, Alan. "How internet companies build software." *MIT Sloan Management Review* 42.2 (2001): 75-84.
- 15) Kuhn, D. Richard, Raghu N. Kacker, and Yu Lei. *Introduction to Combinatorial Testing*. CRC press, 2013.