

A SURVEY ON EFFECTIVENESS OF SOFTWARE TESTING TECHNIQUES AND STRATEGIES

V.Sathyavathy¹, D.Shanmuga Priyaa²

ABSTRACT

Software testing provides an environment to reduce errors, cut maintenance and overall software costs. There are various software development and testing methodologies, testing techniques have emerged over the last few decades to enhance software quality. Software testing is an important activity in the software development life cycle and it is commonly and widely used validation approach in software industry, deployed by programmers and testers. The program with the high complexity cannot be tested completely. More Innovative methods are needed to perform testing as a whole and unit testing with minimum effort and time. Unit testing is mostly done by the developers under a lot of schedule, since the software companies find a compromise between the functionalities, time to market and quality. Thus there is a need for reducing unit testing time by optimizing and automating the process. Test case generation is an error-prone, tedious and time consuming part of unit testing. Two techniques are used to automatically generate the test cases from the input domain using scatter search and search for branch coverage criteria with respect to cyclomatic complexity measure.

Keywords: Software Testing Techniques, Functional Testing, Structural Testing, Manual Testing, Automated Testing

I. INTRODUCTION

Software testing is one of the oldest one as the hills in the history of digital computers. The testing of software is an important factor in assessing the software to determine its quality. Since testing typically consumes 40 - 50% of the total development efforts, and consumes more effort and time for systems that require higher levels of reliability and quality, it is a significant phase of the software engineering. Software testing refers to process of evaluating the software with intention of finding errors in it. Software testing is a technique that aims at evaluating an attribute or capability of a program or product and determining that it meets its quality. Software testing is also used to test various software quality factors like reliability, usability, integrity, security, scalability, performance, efficiency, portability, maintainability, compatibility etc.

II. TYPES OF SOFTWARE TESTING TECHNIQUES

Software Testing Techniques can be divided into the following two types:

¹Ph.D. Research Scholar, Karpagam University

²Professor, Department of Information Technology, Karpagam University

A) Manual testing (static testing)

It is a slow process where testing is done statically. It is done in early phase of life cycle. It is also called static testing. It is done by analyst, the developer and the testing team [1].

The different Manual testing Techniques are:

- 1) Walk through
- 2) Informal Review
- 3) Technical Review
- 4) Inspection

B) Automated Testing (Dynamic testing)

In this tester executes the test script on the testing tool and the testing is done.

Automated testing is also called dynamic testing. Automated testing is classified into four types

- 1) Correctness testing
- 2) Performance testing
- 3) Reliability testing
- 4) Security testing

2.1 Correctness Testing

The most primary purpose of testing is correctness which is the minimum requirement of software. Correctness testing tells about the right behavior of the system from the wrong one for which it will need some type of technique. Either a white box technique or black box technique can be taken in testing the software as a tester may or may not know the details of the software module under test.

2.2 Performance Testing

Performance Testing is the testing that involves all the phases as the mainstream testing life cycle as an independent discipline which involve the strategies such as design phase, design analysis, execution and reporting

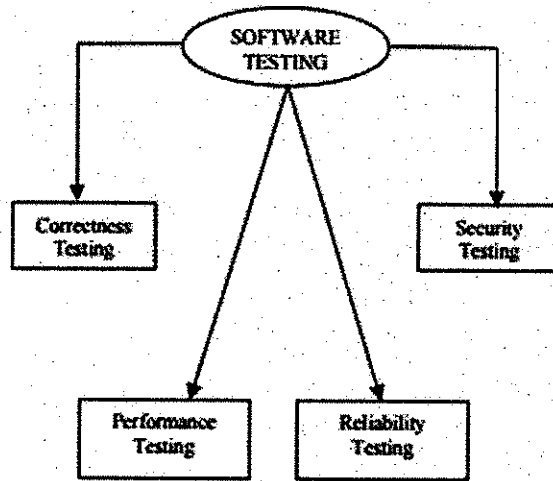


Figure 1. Different forms of software Testing Technique

2.3 Reliability Testing

It is very important testing, as it discovers all the failures of a system and removes them before the system is implemented [4]. This type of testing is related to many aspects of software in which testing process is included; this process is an effective sampling method to measure the software reliability. A model for estimation is prepared in reliability testing which is used to analyze the data to estimate the present and predict future reliability of software.

2.4 Security Testing

It is another important technique that makes sure that only the authorized persons can access the program

and only those can access the functions available to their security level. Security testing is applied to any developed system or (system under development) it is used to find the major loopholes and weaknesses of a system which can cause any damage to the system by an authorized user.

III. PROBLEM OF SELECTING TESTING TECHNIQUES

The evaluation of software testing technique is a very important process, as it is directed to assure the software quality. One of the factors that affect the cost of testing is the number of test cases that are used. If the number of test cases that are generated are more, the longer it will take to plan, execute and analyze the tests. This makes it unworkable to run all possible combinations of input values that rules out exhaustive testing (Myers, 1970). The tests run only on a relatively small set of cases that are chosen from the infinite set of system inputs. The choice of test cases is of very much important, not only because the resulting set should be of minimum size, but also because test set must be on the basis of a small number of inputs, the behavior of the system for the input universe. Suitable testing techniques are used to find a fault with possible set of test cases.

IV. VAUATION OF SOFTWARE TESTING TECHNIQUES

Software testing should be very effective in case to prevent critical damages to the whole system for users, by taking into consideration of failures to the program and its environments. One way to avoid such failures is to choose an exhaustive testing of the system, tests the system with all possible

combinations of inputs which includes both valid and invalid test cases. Excluding the cases, exhaustive testing is an impractical thing for most software systems. It is often noticed that the time and resources, which can limit the ability to effectively complete the testing efforts. It is not mandatory to go for exhaustive testing, rather a testing technique in relation to the selected test strategy that will detect maximum possible critical faults and brings the product to an acceptable level while consuming less resources and time. Whether the testing type is static or dynamic, there is a selection of testing methods [6]. In each testing method there are varieties of testing techniques that can be used to test a system.

Each testing technique is meant for testing its own dimensions i.e. for what purpose it is used, what aspect it will test, what will be its deliverables etc. A different approach to software development requires different testing methodologies and techniques. This limits the ability to use a generic technique for testing a system. At present a wide variety of software testing techniques are used to test a system which will ensure that a variety of defects are found, resulting in more effective testing.

V. WHY TO EVALUATE SOFTWARE TESTING

1. Testing to detect risk: In addition to the number of failures that occur during testing, one must keep track of the cost of those failures. For instance, testing technique A will be considered more effective than testing technique B if the expected total cost of failures detected during test is higher for A than for B

2. Testing and debugging to reduce risk: It is assumed that each failure that occurs during testing leads to the correction of the fault that caused that failure, that reduces the risk associated with the corrected software and the results in the increase of the reliability of software. In case if testing technique A will be considered more effectiveness than testing technique B if A reduces the risk more than B does, thus resulting in less risky software.

3. Testing to estimate risk: To estimate the software reliability, it is assumed that some faults will remain in the software. The objective is to estimate the probability that the software will fail after implementation (during some specified time). Here, it is said that the testing technique A is better than testing technique B for a given technique for estimation of risk if A provides more accurate estimates of risk than B.

VI. SOFTWARE TESTING STRATEGIES

The strategies for software Testing includes software test case design methods into a well planned series of steps that result in successful implementation of software. Software testing strategies provides the road map for testing. A software testing strategy should be strong enough to promote a customized based testing approach ,but at same time it must be right enough. Strategies are generally developed by the project managers, software engineer and testing specialist. There are different software testing strategies.

- 1) Unit testing.
- 2) Integration testing.

3) Acceptance/Validation testing.

4) System testing.

VI. METHODS OF SOFTWARE TESTING TECHNIQUES

Functional testing techniques are used to design the test cases based on the functional requirements of the software product. The goal of functional testing is to choose possible set of inputs according to the specifications of the program to test the program so that all the functions and sub functions within the program are tested at least once. Functional testing uses the boundary value analysis method to identify equivalence classes in the input data. Then the test cases are generated based on the analysis by focusing on equivalence-class boundaries are chosen, run the test cases, and compare the actual result with the expected result to detect failures.

Structural testing techniques are used to plan and design the test cases based on the internal structure of component or system; most commonly internal structure is referred to as the structure of the code. Test cases are designed and generate to get the required coverage for the specified particular item. The main objective of structural testing is to choose a set of inputs according to the structure of the program and aim that all parts of the program (statements, branches or paths) are tested at least once. This type of testing uses branch coverage that analyzes a source-code listing to construct test cases that will lead to 100% branch coverage and statement coverage. After running the tests, the actual output is compared with the expected output to detect the failures.

VII. COMBINING TESTING TECHNIQUES

One thing that was observed in the survey of empirical studies is that the techniques do appear to be finding different types of faults. So an obvious extension of this work is to explore the effectiveness using combinations of techniques [5]. All possible combinations were taken into account. They are:

1. Code Reading and Functional Testing
2. Code Reading and Structural Testing
3. Functional Testing and Structural Testing
4. Code Reading, Functional Testing and Structural Testing

| Techniques | Number of defects detected | Time Taken |
|--------------------|----------------------------|------------|
| Code Reading | 6 | 141 |
| Functional Testing | 8 | 169 |
| Structural Testing | 7 | 154 |

Table 1.1 Number of Defects isolated by each Fault
Table 1.1 shows the number of defects detected and isolated and the time taken by each technique. On an average 63.63% defects were detected and isolated.

Functional testing detected highest number of defects followed by structural technique which was followed by code reading technique. Regarding time, the techniques on average took 86% of the total time taken to detect and isolate the faults in the program.

Code reading was most efficient testing technique followed by structural testing which is followed by functional testing.

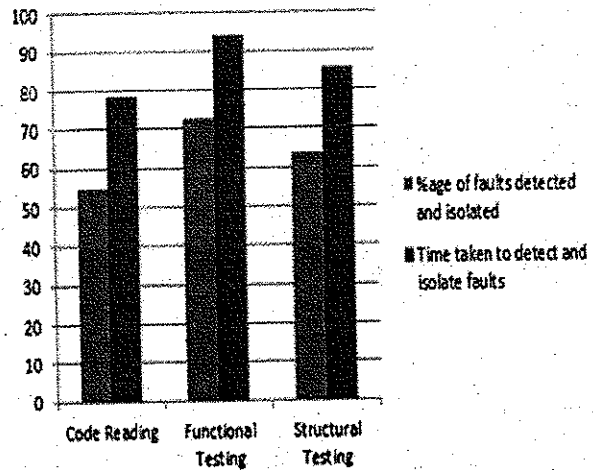


Figure 2. Effectiveness of Each Testing Technique

VIII. CHARACTERISTICS OF A GOOD TESTING TECHNIQUE

Each technique is better for certain things, and not as good for other things. Each individual technique is focused on particular types of defect as well. For example, state transition testing is more effective to find the boundary defects. Testing techniques should ensure the maximum effectiveness with the least possible combination of number of test cases. The "right technique at right time" is the one that lets to achieve the goal, and which the current situation can be accomplished.

However, testers have to answer an important question, which are the best-suited techniques every time they have to test a system. Some techniques are more efficient in finding failures and some are easier to apply than others. Some techniques are more applicable to certain circumstances and test levels; others are applicable to all test levels [9]. Each testing technique that is defined is meant for testing that has its own dimensions i.e. for what purpose it is used, what aspect it will test, what will be its deliverables

etc. It is imperative to find the most effective and efficient testing technique but that should not be practically impossible. Focusing on selection of testing techniques, there are still many decisions to be made about which techniques are the best. Characteristics of good testing technique are: High probability of finding errors to achieve the effectiveness. Probability of finding undiscovered errors achieves its desired goal in the least amount of time and budget, with non-redundant and right level of complexity.

IX. COMPARISON CRITERIA FOR TESTING TECHNIQUES

A basic question is that how the testing techniques can be compared with each other. A general approach is to compare the effectiveness of various software testing techniques. Effectiveness of a testing is a measure of bug finding ability of the testing technique. It can be calculated as Test Effectiveness = $\frac{\text{Errors reported by Testers}}{\text{Total Errors reported}}$ where Total Errors = Tester reported + User reported Errors.

Effectiveness of testing can be adjusted by using several set of parameters which includes Increase in software reliability, Software type, Error detection effectiveness (detection of most errors), Error detection cost (#errors/effort), Error type (Classification of errors found: Critical, Serious, Medium and Low). It is not very easy to compare the effectiveness of the different techniques.

The effectiveness of a technique for testing particular software will, depend on the type of errors that exist

in the particular software. However, based on the nature of the techniques that is being used, one can make some observations about the effectiveness for different types of errors that are present. For comparison, it is best to identify and classify the errors into different categories. Another technique of measuring the effectiveness is to consider the "cost effectiveness" of different strategies, that is, the cost of detecting and isolating an error by using a particular strategy. The cost includes all the effort that is required to plan, test and evaluate.

Classification of techniques is based on the strengths & weaknesses of each technique that is applied (theoretical, technical & pragmatics aspects) would be much more useful than classifications based on the mechanical or operational considerations.

X. EVALUATION RESULTS

- A. There is no clear, and consistent evidence that one fault finding technique is effective than the others, rather than the evidence to date suggests that each technique has its own merits and demerits.
- B. Some studies state that technique A is ranked higher than technique B. Some studies conclude that technique A and technique B find different kinds of defects, and are as such complementary.
- C. The effectiveness of verification activities is very low; only 25-50% of the defects are found by using inspection, where as 30-60% are found using testing.

- D. By combining the testing techniques, defects uncovered are more than by using a single technique.
5. Combining individual testers seem to increase the effectiveness of defect detection that more than combining test case design techniques.
- E. Defect detection effectiveness highly depends on the differences between the individual testers. The variation that arises between the individuals seems to be greater than the variation between techniques as the different testers seem to find clearly different defects in spite of using the same technique.
- F. The effectiveness of defect detection seems to be correlated with the amount of test cases that are used.
- G. The effectiveness of different techniques seems to depend on the type of software that is tested and the types of the actual faults in the software.
- H. It seems that some types of faults are not well suited to some software testing techniques.
- I. There appears some set of relationship between the programs, or the type of defects entered in the programs, and technique's effectiveness.

XII. CONCLUSION

Perhaps the most important thing to be observed is that the best testing technique is no single testing technique to be followed. Different techniques will have different strengths and weaknesses. No technique is good at detecting all types of errors, and

hence testing technique can suffice for proper verification and validation. To be able to evaluate techniques to find out which one is best among them in terms of effectiveness, efficiency and applicability, Experiments are to be conducted on a large scale, under a common standardized framework. But creating a framework for defining, and just exposing techniques is not an easy task in present testing scenario.

REFERENCES

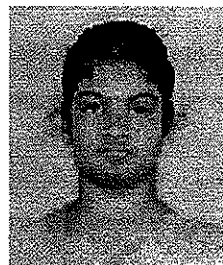
- [1] Chilenski, John Joseph Chilenski and Steven P. Miller, 1994. "Applicability of Modified Condition/Decision Coverage to Software Testing", Software Engineering Journal Vol. 9, No. 5, pp.193-200.
- [2] Edvardson, J. 1999. "A Survey on Automatic Test data generation", In proceedings of the second conference on computer science and engineering Vol.2, No.1, pp.343-351.
- [3] Eugenia Diaz, Javier Tuya, Raquel Blanco, Jose Javier Dolado, 2008. "A tabu search algorithm for structural software testing", Computers and Operations Research Vol. 14, No. 3, pp.38-69.
- [4] Ferguson and Korel, B. 1966. "The chaining approach for software test data generation", ACMTOSEM vol. 5, pp.63-86.
- [5] Howden, W.E. 1977. "Symbolic testing and the DISSECT symbolic evaluation system", IEEE Transactions on Software Engineering vol.3, no. 4, pp. 266-278.

- [6] John Clarke, Mark Harman, Bryan Jones. 2000. "The Application of Metaheuristic Search techniques to Problems in Software Engineering", IEEE Computer Society Press Vol.42, No.1, pp.247-254.
- [7] Boris Beizer 2000. 'Software Testing Techniques', Second edition.
- [8] Abha Jain, Manish Jain, Sunil Dhankar- "A Comparison of RANOREX and QTP Automated Testing Tools and their impact on Software Testing". International Journal of Engineering, Management & Sciences (IJEMS)ISSN-2348 -3733, Volume-1, Issue-1, January 2014 .
- [9] Harpreet Kaur, Dr.Gagan Gupta. "Comparative Study of Automated Testing Tools: Selenium, Quick Test Professional and Testcomplete". IJERA ISSN : 2248-9622, Vol. 3, Issue 5, pp.1739-1743 Sep-Oct 2013.
- [10] Mohd. Ehmer Khan, Farmeena Khan- "A Comparative Study of White Box, Black Box and Grey Box Testing Techniques". (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 3, No.6, 2012.
- [11] H.S. Samra- "A Study on Software Testing"- International Journal of Advanced Research in Computer Science and Software Engineering". IJARCSSE ISSN: 2277 128X Volume 3, Issue 1, January 2013.
- [12] Muhammad Shahid, Suhaimi Ibrahim and Mohd Naz'ri Mahrin- "A Study on Test Coverage in Software Testing"-International

Conference on Telecommunication Technology and Applications-IACSIT Press, Singapore. Proc .of CSIT vol.5 2011.

- [13] Shivkumar Hasmukhrai Trivedi- "Software Testing Techniques"- International Journal of Advanced Research in Computer Science and Software Engineering ISSN: 2277 128X, Volume 2, Issue 10, October 2011.

AUTHORS BIOGRAPHY



V. Sathyavathy is currently working as an Assistant Professor in the department of Computer Technology at KG College of Arts and Science, Coimbatore. She has completed her under-graduation in Bachelor of Science in Computer Technology at Ramakrishna Engineering College, Coimbatore. She has completed her Post Graduation in Computer Application at KGISL-IIM.



Dr.D.Shanmuga Priyaa, working as a Professor in the department of Information Technology at Karpagam University Coimbatore. She had completed Ph.D in Computer Science at Karpagam University in August 2014 and she got 16 yrs of experience. At present she is guiding 8 Ph.D and 1 M.Phil. scholars. She has published more than 15 international journals and more than 2 international conferences.