# A Review of Empirical Evaluation of Software Testing Techniques with Subjects

**Manika Tyagi, Sona Malhotra**

*Abstract*— **Software Testing activities are aimed at detecting errors in the software products and to enhance product quality throughout the entire life cycle of software development. Improper and inadequate testing has resulted many social issues, financial problems and software related problems. In order to carry out testing, using numerous techniques involves excessive use of resources and time. Many techniques has resulted in duplication of effort because they find same type of faults. For this purpose, proper selection of testing techniques is important. So there is need to evaluate testing techniques. Most of the empirical research had conducted to evaluate software testing techniques in terms of effectiveness and efficiency.**

**In this paper, several past empirical studies along with their results are reviewed, conducted with subjects to examine both static (code reading) and dynamic (functional and structural) testing techniques. Finally, the overall performance of testing techniques have been observed with respect to effectiveness and efficiency in existing experiments, and it can be concluded that functional (black box) testing is most effective and efficient and code reading is least effective and efficient.**

*Index Terms*—**code reading, empirical study, evaluation, experimentation, functional testing, structural testing**

## I. INTRODUCTION

Software testing is the essential phase of software development life cycle. while developing software, software testing costs between 40% to 80% of the total cost of development of software. The main objective of testing is to detect faults and failures that occurred during development and to ensure that software is bug-free. Improper and inadequate testing has resulted many social issues, financial problems and software related problems. To test the system exhaustively is one such solution, but with limited resources, time and money, it is not practical. For this purpose, proper selection of testing techniques is important, So it is likely to choose effective testing techniques.

To get enough information about how effectively they do? How much resource they utilize, and depend on parameters that they have taken into consideration, is difficult. In order to carry out testing, using numerous techniques, involves excessive use of resources and time. Many techniques have resulted in duplication of efforts because they find same type of faults. Therefore, there is need to evaluate testing techniques.

Most of the studies conducted to evaluate software testing techniques. Juristo et al. [18] categorized these studies into theoretical studies, empirical studies with subject and empirical studies without subjects. These studies could be categorized as: analytical studies, empirical studies and theoretical studies.

Theoretical studies aim to examine unadulterated techniques from angle of logic and based on deductive reasoning. The techniques based on their theoretical groundwork are examined and are highly useful to enlarge our knowledge behind testing techniques. They analyzing the effectiveness of code based [8], [11] or regression testing [12], [13]

Empirical studies include controlled experiments to evaluate software testing techniques. The solution of empirical studies would be based on practitioner's mindset and extensive studies of effectiveness of several testing techniques in practice. Empirical studies can be performed with subjects and without subject and based on inductive reasoning and logic. Empirical studies without subjects examine test-case generation and compare efficiency and effectiveness of different testing techniques [9], [10]. Some studies examine approaches for selection of test-case [14], [15]. Empirical studies with subjects is a simulation of real situation. It takes into account how the subject influence technique behaviour. Most of the studies conducted to evaluate static and dynamic testing techniques in terms of efficiency and effectiveness [1], [2], [3], [4], [5], [6], [7].

Analytical studies resembles with theoretical studies in nature and produce generalized results, the results which are applicable to any experimental perspective. The conclusions of analytical comparisons are based on statistical terms.

## II. SOFTWARE TESTING TECHNIQUES CLASSIFICATION

Software testing techniques are classified into two broad categories: Static testing and Dynamic testing.

### *A. Static Testing Techniques:*

Static testing techniques focus on testing the software product without the actual execution of source code of software .It covers the analysis and checking of system representations such as requirement document, design and source code of system without executing it, either manually or automatically.

Automatic testing focus on testing program or program related documents by using software tools. In static automatic

 All Rights Reserved © 2014 IJARCET 

testing, static analysis tool is used for testing program, the source code is input into this tool and evaluated for quality.

Manual static testing focus on testing the program and program related document without using any tool.

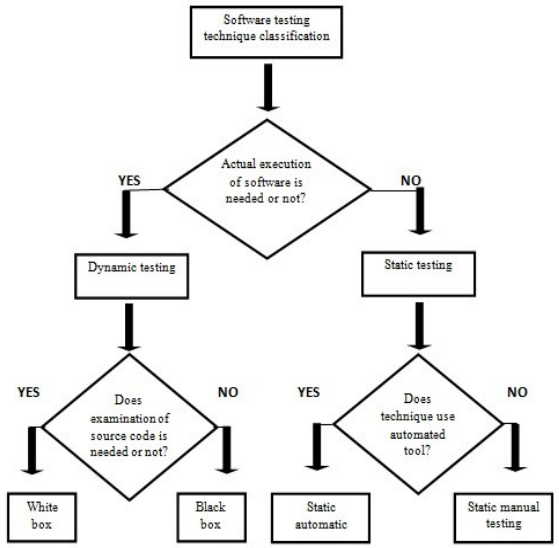


**Fig. 1** Software testing technique classification

*B. Dynamic Testing Techniques*

Dynamic testing techniques focus on testing software product through the actual execution of software. The software product is tested in real or simulated environments, for both normal and abnormal inputs, in order to check how the system respond to different inputs data sets. A system is dynamically tested means execution, and by studying the result of execution, the quality levels set for dynamic evaluation can be decided.

Dynamic testing techniques are classified into two categories: White Box testing and Black Box testing

**White Box Testing**

In white box testing, the knowledge of source code is required for designing test cases. It concentrates on logic and internal structure of program code, and concerned about requirements of software which is under test. It is also known as structural testing. It mainly deals with examining the logic of program or software product. In White box testing techniques, test cases incorporate coverage of code written in terms of branches, conditions, statements, and internal logic of program code etc.

To implement white box testing strategy, knowledge of coding and logic of software systems must be necessary and results are evaluated based on a set of coverage criteria. There are several types of white box testing, but only those that were evaluated in the mentioned past empirical studies, have discussed.

Statement Coverage: It requires that test cases to be designed so as to execute and test each statement in a program at least once.

Branch Coverage: It aims to design test cases to make each branch condition in the program to assume true and false value in turn.

Condition Coverage: It aims to design test cases so as to make each component of a composite conditional expression to assume true and false value in turn.

Loop Testing: It aims to design test cases so as to continuously execute the loop until the condition become false and testing whether it is proper or not.

Path testing: It aims to design test cases so as to execute and test all basis paths in the program at least once.

**Black Box Testing**

In Black Box testing, no knowledge of source code is required, test cases are designed from examination of input/output values only. Black Box testing reflects only behavior of software system, it focus on what the system perform. It is also known as functional testing. To implement black box testing strategy, knowledge of functional specification of system must be necessary, so that all functions of system are tested at least once. Black Box Testing mainly focus on testing functionalities and requirements of system. Black box testing can be categorized as follows: Equivalence partitioning and boundary value analysis.

Equivalence partitioning: Identify the equivalence classes for program and test cases are generated for each equivalence class identified.

Boundary value analysis: It aims to design test cases using the values at boundaries of equivalence class identified

The overall process of evaluation of test design technique is represented in Fig.2.is described in Eldh et al. [17]. Prepare faulty program by injecting faults in unadulterated program. The experiment conducted with subjects, they applied different testing techniques to faulty program. The next step consists of evaluation of testing techniques, based on results achieved.

All Rights Reserved © 2013 IJARCET

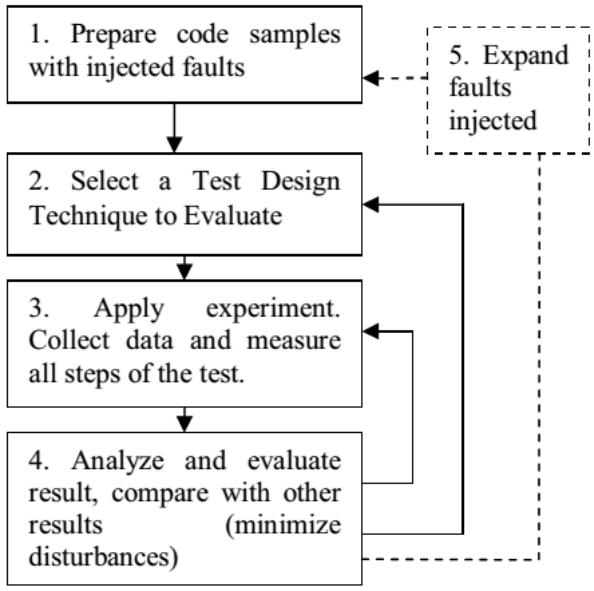


**Fig.2** Overall process of evaluating test design techniques

## III. PREVIOUS STUDIES

Hetzel, W. [1] conducted an experiment with 39 subjects for comparing effectiveness of three testing techniques i.e. functional testing, code reading and structural testing. The experiment was based on testing three program coded in PL/I. The result of experiment was that functional testing and structural testing was equal in effectiveness, while code reading was less effective.

| Aspect | Results of experiment |
|---|---|
| Effectiveness (Detection) | -Subjects who applied the testing technique performed more effectively than those who applied the reading technique.<br>-Among the two testing techniques there was no significant difference in effectiveness |

**Fig. 3** Hetzel's results on effectiveness of testing techniques

Myers [2] performed a controlled experiment with 59 subjects (professional programmers) to compare efficiency and effectiveness of static testing techniques(walk-through /inspection) and dynamic testing techniques(functional testing and structural testing).The experiment was based on testing single program which was coded in PL/I with 63 lines of code(LOC) .The result of the experiment, in terms of efficiency, was that structural testing required least time, functional testing consumed less time and walk-through /inspection method consumed the most time. The result of experiment, in terms of effectiveness was that all three techniques were equally effective.

| Aspect | Results of experiment |
|---|---|
| Effectiveness (Detection) | -Among the three testing techniques there was no significant difference in effectiveness. |
| Efficiency (Detection) | -Structural testing required least time, functional testing somewhat less amount of time, walkthrough/ inspection required the most time. |

**Fig. 4** Myers's result on effectiveness and efficiency of testing techniques.

Selby [3] to compute the effectiveness and efficiency of three testing techniques. They extended the experiment by including a fault isolation phase after fault detection phase.

| Aspect | Results of experiment |
|---|---|
| Effectiveness (Detection) | -Depends upon program, not on technique. |
| Effectiveness (isolation) | -Depends upon subject and program, not on technique. |
| Efficiency (Detection) | 1) Condition coverage takes more time than boundary value analysis.<br>2) Time spent on finding faults also depends on subjects.<br>3) Condition coverage has lower fault rate than boundary value analysis. |
| Efficiency (isolation) | 1) Depends upon subject and program, not on technique.<br>2) For inexperienced subjects: boundary value analysis takes longer than Condition coverage. |
| Efficiency (total) | 1) For inexperienced subjects: Condition coverage takes more time than boundary value analysis<br>2) Time also depends on subject. |
| Fault type | -For both isolated and detected: there is no difference between techniques. |

**Fig.5** Kamsties and Lott 's result on effectiveness and

. The experiment was based on testing three programs which was coded in C. The testing techniques were code reading by stepwise abstraction, functional testing and structural testing (100% branch coverage, multiple condition and relational operator coverage). They performed two replication of experiment, both involved subjects. There was no significance correlation between replication-I and replication –II, both conducted at different durations. Replication-I had conducted during summer semester with 27 subjects. Replication-II had conducted with 23 subjects during winter semester. The result of the experiment was that functional testers were most efficient in observing failures and isolating

 All Rights Reserved © 2014 IJARCET 

faults. In either of the replication, all the three testing techniques were almost same in effectiveness.

Roper et al. [5] replicated the experiment of Kamsties and Lott [4]. The experiment had conducted with 47 subjects to evaluate the effectiveness of testing techniques and combination of techniques. The testing techniques were functional testing using boundary value analysis, structural testing using branch coverage and code reading by stepwise abstraction. The experiment used three program coded in C, to which testing techniques were applied. The result of the experiment was that effectiveness of techniques was dependent on program to which testing techniques were applied, and on the type of faults.

| Aspect | Results of experiment |
|---|---|
| Effectiveness (Detection) | -Depends on the technique/program combination.<br>-Depends on nature of faults. |
| Combination of techniques | -Higher number of faults combining techniques. |

**Fig.6** Roper's result on the effectiveness of testing techniques and combination of techniques

Juristo and Vegas [6] replicated the experiment of Roper et al. [5] to evaluate the effectiveness of static testing techniques(code reading by stepwise abstraction) and dynamic testing techniques(functional testing using equivalence class partitioning and structural testing using branch coverage). The experiment was based on testing four program coded in C. They performed two replication of their experiment. In replication-I and replication-II the experiment was conducted with 195 subjects and 46 subjects respectively. The result of the experiment was that effectiveness of techniques depends on program, techniques and fault type. Functional testing and structural testing behave identically with respect to fault type. Code reading behave worse.

| Aspect | Results of experiment |
|---|---|
| Effectiveness (Detected and observable) | 1) Depends on technique, program and fault.<br>2) Code reading behaves worst than functional testing and structural testing, indistinctly for the defect type. With regard to functional testing and structural testing, both behaves identically.<br>The number of subjects that detect a defect influence the program version |

**Fig.7** Juristo and Vegas's results on effectiveness of testing techniques

Farooq et al. [7] replicated the experiment of Kamsties and Lott [4] to compare software testing techniques i.e. static testing techniques(code reading) and dynamic testing techniques(functional testing and structural testing). The experiment was conducted with 18 subjects and was based on testing three programs coded in C. The testing techniques has evaluated in terms of effectiveness, and efficiency. Effectiveness has measured in terms of number of faults detected and isolated. Efficiency has measured in terms of time required to detect and isolate faults. The result of the experiment was that effectiveness depends on program. Efficiency depends on program and techniques.

| Aspect | Results of experiment |
|---|---|
| Effectiveness (Detection) | Depends upon program, not on technique. |
| Effectiveness (isolation) | Depends upon program |
| Efficiency (detection) | Depends on program |
| Efficiency (isolation) | Depends on technique |

**Fig.8** Farooq and Quadri's result on effectiveness and efficiency of testing techniques

The results of above experiments, presented in this section are discussed in Juristo et al.[16].

## IV. RESULTS

In this section, the main results from above experiments are listed. From fig. 9, it can be concluded that functional testing is most effective, structural testing is less effective and code reading is least effective, in almost all experiments i.e.

FT > ST > CR (with respect to effectiveness).

FR- functional testing

ST- structural testing

CR- code reading

All Rights Reserved © 2013 IJARCET

| AUTHOR | Black box testing | White box testing | Code reading |
|---|---|---|---|
| Hetzel | High | Medium | Low |
| Myers | Low | Medium | High |
| Basili & Selby | High | Low | Medium |
| Kamsties & Lott<br>Replication1<br>Replication2 | <br>High<br>High | <br>Medium<br>Medium | <br>Low<br>Medium |
| Roper et.al | medium | High | Low |
| Juristo and Vegas<br>Replication 1<br>Replication 2 | <br>High<br>High | <br>Medium<br>Medium | <br>Low<br>× |
| Farooq | depends on program | depends on program | depend on program |

**Fig. 9** overall Performance of testing techniques with respect to effectiveness in existing experiments and × denotes that not considered in corresponding experiment.

From fig. 10, it can be concluded that functional testing is most efficient, structural testing is less efficient and code reading is least efficient, in most of the experiments i.e. FT>ST>CR (with respect to efficiency).

| AUTHOR | Black box testing | White box testing | Code reading |
|---|---|---|---|
| Hetzel | × | × | × |
| Myers | Medium | High | Low |
| Basili & Selby | depends on program | depends on program | depends on program |
| Kamsties & Lott<br>Replication 1<br>Replication 2 | <br>High<br>High | <br>Medium<br>Medium | <br>Low<br>Low |
| Roper et.al | High | Medium | Low |
| Juristo & Vegas<br>Replication 1<br>Replication 2 | <br>×<br>× | <br>×<br>× | <br>×<br>× |
| Farooq<br>(fault detection)<br><br>(fault isolation) | <br>depends on program<br>depends on technique | <br>depends on program<br>depends on technique | <br>depends on program<br>depends on technique |

**Fig.10** Overall Performance of testing techniques with respect to efficiency in existing experiments and × denotes

## V. CONCLUSIONS

Several past empirical studies are reviewed conducted with subjects, to evaluate the effectiveness and efficiency of testing techniques. The techniques includes both static (code reading by stepwise abstraction) and dynamic (functional and structural) testing. Functional testing includes boundary value analysis and equivalence partitioning and Structural testing (statement coverage, branch coverage condition coverage, loop and relational operator coverage) were evaluated. Finally we concluded that functional testing is most effective and efficient, structural testing is less effective and efficient and code reading is least effective and efficient. In other words, static testing is less effective and efficient than dynamic testing.

.

## REFERENCES

[1] W. C. Hetzel, "An experimental analysis of program verification methods," Phd thesis, University of North Carolina at Chapel hill, 1976.
[2] G. J. Myers, "A controlled experiment in program and code walkthrough/ inspection," Communication of ACM, 21(9):760-768, September 1978.
[3] V. Basili and R. Selby, "Comparing the effectiveness of software testing strategies. Software Engineering," IEEE Transcation on (12):1278-1296, 1987.
[4] E. Kamsties, and C. Lott, "An empirical evaluation of three defect detection techniques," software Engineering ESEC95, pages 362-383, 1995.
[5] M. Roper, M. Wood, and J. Miller, "An empirical evaluation of defect detection techniques," Information and Software Technology, 39(11): 736-775, 1997
[6] N. Juristo and S. Vegas, "Functional testing, Structural testing and Code Reading: what fault type do they each detected?," Empirical Methods and Studies in Software Engineering Pages 208-232, 2003
[7] S. U. Farooq, S. M. K. Quadri and N. Ahmad, "A controlled experiment to evaluate effectiveness and efficiency of three software testing methods," IEEE Conference on Software Testing, Verification and Validation, 2013
[8] T. Y. Chen, Y .T. Yu, "On the relationship between partition and random testing,". IEEE Transaction on Software Engineering, 20(12): 977-980, 1994.
[9] A. J. Offutt, S. D. Lee, "An empirical evaluation of weak mutation," IEEE Transactio on Software Engineering, 20(5):337-344, 1994
[10] A. J. Offutt, A. Lee, G. Rothermel, RH. Untch, C. Zapf, "An experimentation determination of sufficient mutant operators," ACM Transaction on Software Engineering and Methodology, 5(2): 99-118. 1996.
[11] L.A. Clarke, A. Podgurski, D.J. Richardson, S.J. Zeil. A formal evaluation of data flow path selection criteria. IEEE Transaction on Software engineering. 15(11): 1318 -1332, 1989.
[12] D. S. Rosenblum, E. J. Weyuker, "Using coverage information to predict the cost effectiveness of regression testing strategies," IEEE Transaction on Software Engineering. 23(3): 146-156, 1997.
[13] G. Rothermel, M. J. Harrold, "Analyzing regression test selection techniques," IEEE Transaction on Software Engineering. 22(8): 529-551, 1996.
[14] G. Rothermel, M. J. Harrold, "Empirical studies of safe regression test selection technique," IEEE Transaction on Software Engineering. 24(6): 401-419, 1998.
[15] T. L. Graves, M. J. Harrold, J. Kim, A. Porter, G. Rothermel, "An empirical study of regression test selection techniques," ACM Transaction on Software Enginering and Methodology, 10(2):184-208, 2001.
[16] N. Juristo, A. Moreno, and S. Vegas, "Reviewing 25 years of testing techniques experiments," Empirical Software Engineering, 9(1): 7-44, 2004.
[17] S. Eldh, H. Hansson, S. Punnekkat, A. Petterson and D. Sundmark, "A framework for comparing efficiency, effectiveness and applicability of Software testing techniques," In testing: Academic and industrial conference –Practice and research technique,2006, TAIC PART 2006.proceedings, Pages 159-170, IEEE, 2006.
[18] N. Juristo, S. Vegas, M. Solari, S. Abrahao and I. Ramos,"Comparing the effectiveness of equivalence partitioning, branch testing and code reading by stepwise abstraction applied by subjects," IEEE Conference on Software Testing, Verification and Validation, 2012.

 All Rights Reserved © 2014 IJARCET 