Implementation of New Approaches of Software Testing Methodology for Finding Error and its Analysis Techniques

[1]Komal Alwani, [2]Dr. Pushpneel Verma

[1]Research Scholar, Deptt. of Computer Application, Bhagwant University, Ajmer, Rajasthan

[2]Associate Professor, Bhagwant University, Ajmer, Rajasthan

ABSTRACT

Software testing is an important step in SDLC. This step not only helps in checking the quality of software. This process of testing is divided into 2 categories, one is model based technology and the other code is driven testing. Code driven test testing is based on complete code (each and every line). Software quality along with software testing is a significant stage of software testing. Software testing is an main tool for assessing software to determine its quality. In view of the fact that testing typically consumes 40 ~ 50% of development effort, and takes more effort for systems that require a higher level of reliability, it is an important part of software engineering. Software testing is also used to test software and to detect other software errors. Software testing is an activity that aims to evaluate the quality of a program and also identify and improve defects. Software testing is the process of uncovering requirements, design and coding errors in a program. It is used to identify the purity, completeness, security, and quality of software products against a specification. Software testing identifies faults, defects or inaccuracies that must be determined. Software testing is an activity intended to evaluate a program's specialty or capability and ensure that it meets the required result. There are several strategies to software testing, but efficient testing of a complex outcome is essentially a process of testing, not simply a topic of creating and following a routeing process. It is usually impossible to detect all errors in the program. Thus this fundamental problem in testing throws open questions, as to what strategy we should adopt for testing. Thus, choosing the right strategy at the right time will make software testing efficient and effective. Software testing is a set of activities intended to find errors in software. The main objective of testing can be quality assurance, reliability estimation, verification or validation. In this thesis we have described software testing methods which are classified by theory.

Keywords: *Software testing, SDLC, Specification, Quality, Complex product etc.*

1. INTRODUCTION

Software error analysis includes techniques used to detect, analyze, and estimate errors related to errors. This includes error detection techniques, single errors analysis, data collection, metrics, statistical process control techniques, error prediction models and reliability models Incorrect detection procedures Improve programming, Programming Quality Confirmation (SQA), Programming Confirmation, Approval, and Testing are used to isolate discrepancies in programming objects. When an inequality is identified, an examination is performed to determine whether the discrepancy is a factual mistake, and to properly distinguish the nature and cause of the defect, assuming that this is the case. It is properly organized. Can. Regularly, the accent is on illuminating a solitary mistake. Despite this, a solitary mistake may be a depiction of individual errors that have arisen from comparable misconceptions, or it may show the presence of significant issues in the advancement cycle. Simply amending a fumble and not having an inclination to fundamental issues can bring additional complications later in the survival cycle. Intensive explosion investigations include classification of fault information, use of lattice and factual cycle control (SPC) methods. The measure is used to conduct a direct survey of an item or cycle, while SPC procedures are used to identify patterns in the advancement cycle and for serious item issues. Inaccurate information on the whole theory can be collected and put into a hierarchical information base for current use or future use. For example, SPC methods can show that a large number of blunders are identified with the plan, and after additional ventures, it is revealed that many manufacturers are making comparative mistakes. It can then be argued that planning philosophy is unprepared for specific technology, or that the creators have not been satisfactorily prepared. Then appropriate changes can be made for the reform cycle, which are useful for the present proposal as well as future adjournment. The classification of the bang information additionally uses the dependency model to assess the probability that the framework will operate for a specific time in a predefined climate without frustration frustration. A trader may use programming dependency assessment strategies to make changes in the testing cycle, and the customer can use these methods in choosing which items to accept. The blunder information collected by a vendor can be valuable to reviewers. Inspectors may demand that vendors furnish incorrect information, although with the understanding that confidentiality will be maintained and will not be resumed. Information collected from vendors can be used by examiners to establish the basis of information, assessing high honesty programming gives a pattern for examination. Information from the previous proposal would control the evaluators of what types of errors they could detect, or the various properties identified by mistakes. For example, it is to be resolved that the motion of thunder under evaluation is contrary to a satisfactory reach and is the previous principle.

Software testing is a set of activities intended to find errors in software. It also verifies and verifies whether the program is working correctly without any bugs. It analyzes software to find bugs. Software testing is not only used to find and fix bugs, but also ensure that the system is working according to specifications. Software testing is a series of procedures designed to ensure that computer code does what it was designed to do. Software testing is a destructive process of trying to find errors. The main objective of testing can be quality assurance, reliability estimation, verification or validation.

Some software tests are included. [6] [7] [8]

- Tested The better it works, the better it can be tested.
- Better software can be controlled More testing can be automated and optimized.
- Ion less change, less interruptions for testing.
- Und a successful test is one that uncovers an undiscovered error.
- Process testing is a process of identifying the correctness and completeness of software.
- The general purpose of software testing is to confirm the quality of the software system by using the software systematically under carefully controlled conditions.

The Classified can be divided into classified [4] by software testing

1. Perfect test
2. Performance Testing
3. Reliability test
4. Safety test

2. LITERATURE REVIEW

The Debugging Process Model (Before 1956): During that period, the concept of program checkout, debugging and testing were not clearly distinguishable. They utilized testing and troubleshooting conversely. Around then, Alan Turing composed two articles and addresses a few inquiries and furthermore characterized an operational test for smart conduct.

The Demonstration Process Model (1957-78): During that period, the investigating testing were plainly recognizable by including endeavors to distinguish, find, recognize and right issue. Charles pastry specialist accentuation on program checkout with two objectives: ensure the program runs, and program takes care of the issue.

The Destruction Process Model (1979-82): Myers composed the book 'The Art of Software Testing', examined programming examination and audit testing strategy. The product testing was first time depicted as "the way toward executing a program with the expectation of discovering blunders".

The Evaluation Process Model (1983-87): The Institute for Computer Sciences and Technology of the National Bureau of Standards distributed Guideline, explicitly focused at government data preparing system(FIPS) for Validation, Verification, and Testing of Computer Software in 1983, in which a strategy that consolidate investigation, audit, and test exercises to give item assessment during the product lifecycle was depicted. The rules guaranteed that a deliberately picked set of VV&T strategies can assist with guaranteeing the turn of events and support of value programming.

The Prevention Process Model (Since 1988): Beizer composed the book 'Programming Testing Techniques' which have most complete inventory of testing methods, and characterized that "the demonstration of planning tests is one of the best bug preventers known."

Manpreet Kaur and Rupinder Singh (2014), Software testing is imperative to diminish mistakes, upkeep and generally speaking programming costs. One of the serious issues in programming testing zone is the means by which to get an appropriate arrangement of experiments to test a product framework. We recognize various ideas that each product designing understudy and staff ought to have learned. There are presently many testing strategies accessible for producing experiments. This set ought to guarantee greatest adequacy with the most un-conceivable number of experiments. The primary objective of this paper is to broke down and contrast the testing procedure with discover the best one to discover the mistake from the product.

Research objectives and approaches

1. To study the reliability test, it detects all the failure of the system and removes them before deploying the system.
2. To find the load test it is to check whether the system can perform well for the specified user.
3. Less change, less interruptions for testing.
4. A successful test is one that exposes an undiscovered error.
5. To find comparative study of software testing
6. A successful examination is one that exposes a hidden error.
7. Testing is a method of identifying the accuracy of software.

3. METHODOLOGY

Software testing is a technique that is used to evaluate the quality and finding error of software developed. It is also a way of finding errors in software and makes it a possible task. It is a functional way of running programs, in which errors are finding.

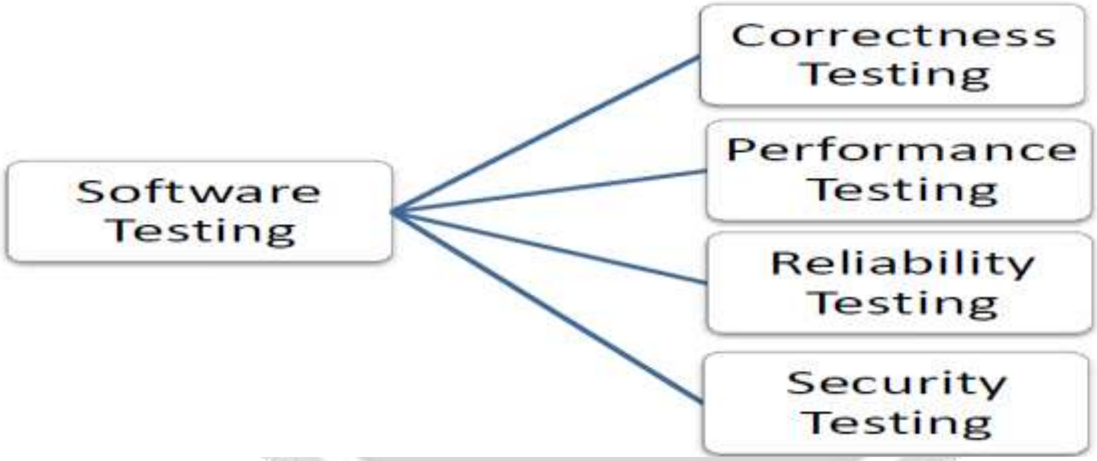

Figure 1.1: Different software testing for finding errors

The most important reason for testing is accuracy, precision and finding errors, which is also the minimum liability of the software testing for error finds. The accuracy of the test indicates the correct performance of the system from the wrong for which it would require some kind of vision.

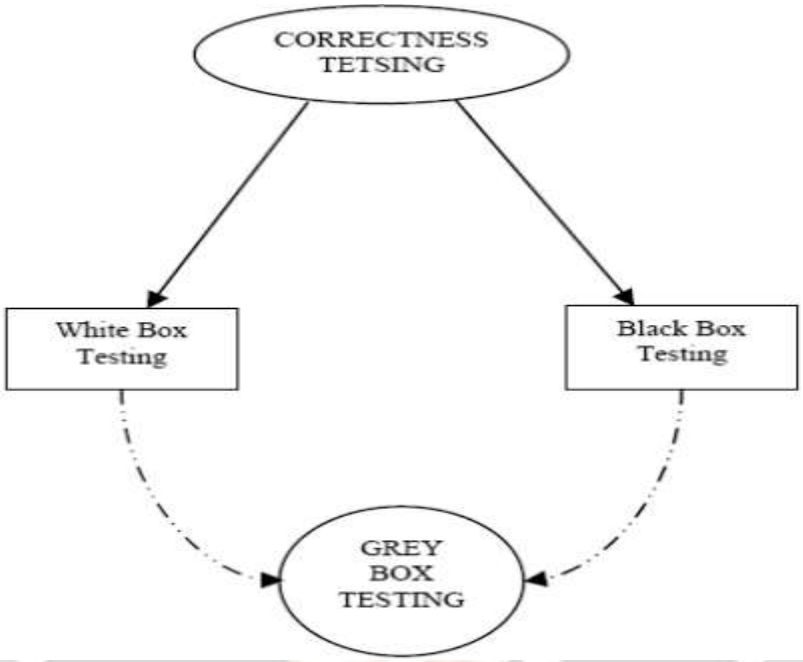

Figure 1.2: Flow chart of Correctness Testing for finding errors

4. RESULT AND DISCUSSION

The objective can be divided into classified [4] by software testing

1. Perfect test
2. Performance Testing
3. Reliability test
4. Safety test

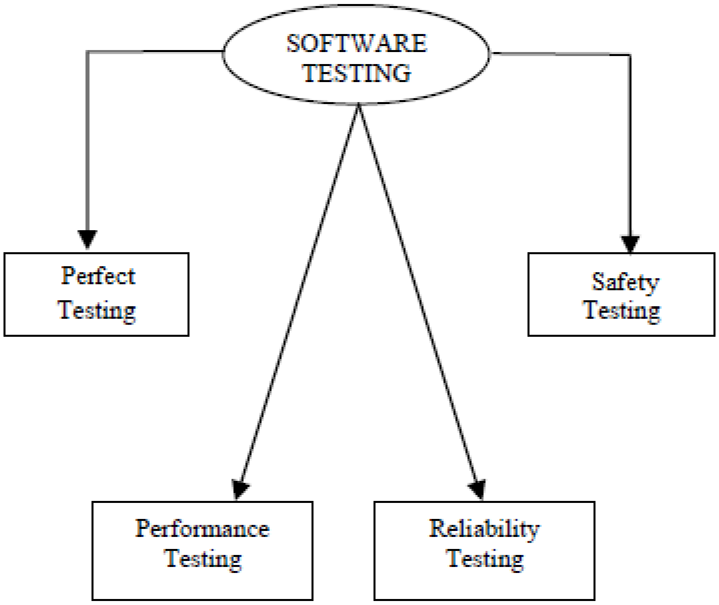

Figure 1.3: Different types of software testing techniques

1. Perfect test

The most essential objective of testing is accuracy, which is also the minimum requirement of software. The correction test states the correct behavior of the system from incorrect to which it would require some sort of Oracle. Either a white box point or a black box point can be taken into the test software because the details inside the software module under test may or may not be known. E.g. Data flow, control flow etc. Considerations of white box, black box or gray box testing are not limited to purity testing alone.

2. Performance Testing

Performance testing encompasses all phases as an independent discipline with a mainstream test life cycle that includes strategies such as planning, design, execution, analysis, and reporting. This test is directed to assess the consistence of a framework or part with a predetermined exhibition necessity. [2] The presentation assessment of any product framework incorporates asset usage, throughput and boost reaction time. By execution testing we can gauge the exhibition attributes of any application. One of the most significant goals of execution testing is to keep up high inactivity, high stream and low utilization of the site. These are two types 1. Load Testing 2. Stress Testing

3. Reliability test

Reliability relates to several aspects of testing software including the testing process; This test procedure is an effective sampling method for measuring software reliability. The robustness test and stress test are versions of the reliability test. By robustness we mean how the software component works in stressful environmental conditions. Robustness testing only addresses robustness problems such as machine crashes, abnormal terminations, etc. Robustness testing is very portable and scalable.

4. Safety test

Safety testing is very helpful for the tester to detect and fix problems. This ensures that the system will run for a time without any major problems. It also ensures that the systems used by any organization are protected from any unauthorized attack. In this way, security testing is beneficial for the organization in all aspects. The five major concepts covered by security testing are confidentiality, integrity, authentication, availability, and authorization.

4.1 Experimental Results

Under the hood, the MC attaches the Nutrition States machine (FSM) to syntactic program objects (eg, variables, memory locations, etc.) and uses interprocessural data ov analysis to calculate the recoverability of the error state. Because conservative pointer analysis is often a source of false positives for bug nading purposes [10], MCs choose not to model bus pointers or heaps, preventing false positives from relationships with false aliases. MC checkers use heuristics (eg, separate FSM transitions for true and false branches of relevant statements) and statistical methods to estimate some lost information. These techniques typically dramatically reduce false positive rates after several rounds of trial and error. However, they cannot fully compensate for information lost during analysis. For example, in the code below,

```
/* 1: data correlation */
if (x) spin lock(&lock);
if (x) spin unlock(&lock);
/* 2: aliasing */
l = &p>lock;
spin lock(&p>lock);
spin lock(l);
```

Application (KLOC)	Err Msgs	Bugs	(FP)	Warn
News Pro (6.5)	9	9	(0)	9
myBloggie (9.2)	15	15	(0)	25
PHP Webthings (38.3)	20	20	(0)	8
DCP Portal (121)	40	40	(0)	60
e107 (126)	14	14	(0)	24
Total	99	99	(0)	126

Table 1.1: experimental Summary

The idea that redundant operations, like type errors, are usually serious accuracy errors. Experimentally this idea is validated by writing and applying V redundancy checkers in real code. The errors often found shocked us. While harmless, the performance of that redundancy is strongly correlated with the presence of conventional hard errors. Demonstrating how redundancy provides a way to detect dangerously specific imaginations. Error detection is an important step in the dependency life-cycle of an application because it helps to find the time point at which an application fails. Error detection methods provide reliability to the application and are the basis for any post-detection diagnosis.

To evaluate each of our consistency checks, we define the accuracy as the ratio of the number of correctly finding errors to injection errors. The ratio of the number of accurately identified errors is defined as the ratio of the number of total known errors. For application generic probes, we give accuracy and precision over the accuracy of an EJB request. For our application-specific checks, we give precision and precision over the specifics of a web request, as they are evaluated at the end of each web request.

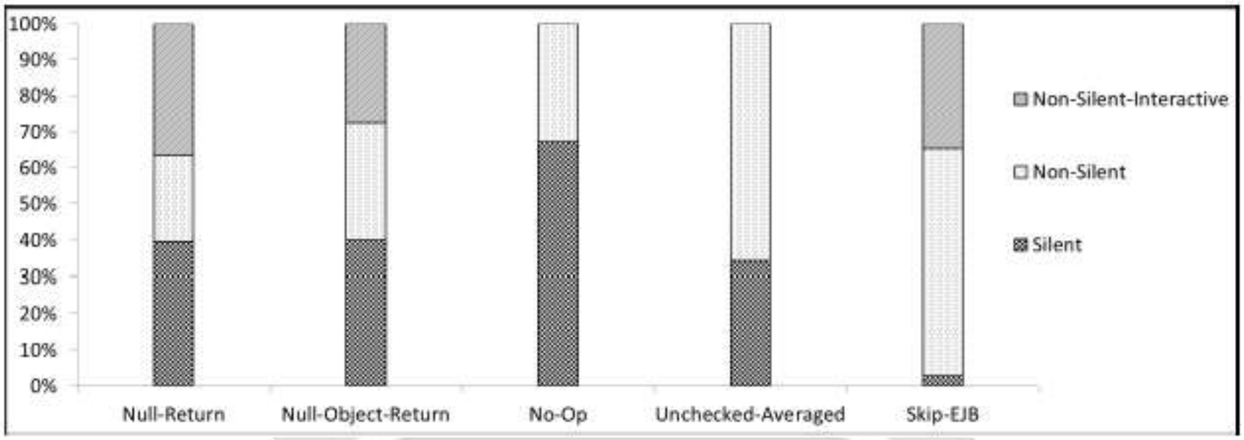

Figure 1.4: Failure Class

CONCLUSION

This paper describes some of the most prevalent and commonly used strategies of software testing. They are 1. Prefect testing 2. Performance testing 3. Reliability testing 4. Safety testing. To illustrate how intercontinental the accuracy of our analysis is, we changed the function summaries and repeated our experiment, which is the smallest codebase. This time, the analysis yielded 19 error messages (as opposed to interprocessural analysis with 8). Upon inspection, all 11 additional reports are false positives due to the user-defined sanitization operation. We discuss how traditional checking approaches, based on annotations or specifications, can use redundancy checking as a safety net for ND missing annotations or incomplete specifications. Typically, such specific citation maps map to redundant functions. For example, suppose we have a specific quotation that binds shared variables to locks. Missed bindings will lead to redundancy: a critical clause that has no shared states and locks that protect without any variables. We can make such defaults because we know that each lock must protect some shared variable and each key clause must have some shared state. Silent failures have a significant percentage of injected errors. Both internal and external mechanisms would be very valuable for the application to log possible expressions for the application and to convert some silent errors to non-silent. Successful use of these techniques in industrial software development will validate research results and advance future research.

References

[1]. F. C. Eigler, V. Prasad, W. Cohen, H. Nguyen, M. Hunt, J. Keniston, and B. Chen, “Architecture of systemtap: a linux trace/probe tool,” 2005.

[2]. “DayTrader - a sample Java EE benchmark application,” https://cwiki.apache.org/GMOxDOC20/daytrader.html.

[3]. J. Duraes and H. Madeira, “Emulation of software faults: A field data study and a practical approach,” Software Engineering, IEEE Transactions on, vol. 32, no. 11, pp. 849 –867, 2006.

[4]. “GlassFish - an application server for Java EE platform,” http://glassfish.java. net.

[5]. “Jboss.” [Online]. Available: http://www.jboss.org/

[6]. Software Testing by Cognizant Technology Solution.

[7]. Introduction to software testing available at http://www.onestoptetsing.com/introduction/

[8]. Software testing techniques available at http://pesona.mmu.edu.my/~wruslan/SE3/Readings/GB1/pdf/ch14-GB1

[9]. Paper by Lu Luo available at http://www.cs.cmu.edu/~luluo/Courses/17939Report.pdf

[10]. Security testing-wikipedia the free encyclopedia available at http://en.wikipedia.org/wiki/securitytetsing.

[11]. White box testing from wikipedia, the free encyclopedia.

[12]. Software testing for wikipedia available at http://en.wikipedia.org/wiki/grey_box_testing#grey_box_tetsing