ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 2, February 2014

Implementing Software Testing Model Approach for Efficient Bug Finding with Yin-Yang Testing Theory on Java Application

Geetika Gandhi[1], Sushil Garg[2]

Assistant Professor, Department of Computer Science, RIMT –IET, India[1]

Professor, Department of Computer Science, RIMT –MAEC, India[2]

Abstract: Model Selection for Software testing is very important prospective in various product accuracy. Through research on software testing model selection, seeking the most appropriate testing method to achieve most reasonable testing volume and optimal testing result. In our Research we will focus on improving this model description by adding more user end experience in acceptance testing. Various experiences from industrial companies will be fetched and will be implemented according to the optimized Yin-Yan Theory for software testing. We will also try to introduce more testing dependencies. In User testing, we will add offline and online dependencies test which will be helpful in finding issues in overall acceptance testing phase. Software testing will be done by various tools and will link to proposed theory. Test model selection and test volume evaluation method will be applied to the software testing work of Industrial applications and will compared with traditional method.

Keywords: Yin-Yang Theory, Regression Testing, Software Testing.

I. INTRODUCTION

Software testing is a set of activities conducted with the intention of finding bugs in software. It verifies and validates whether the program is working as it was meant to with no bugs or not. [3] It analyzes the software. Software testing is not just used for finding and fixing of errors but it also ensures that the system is working in accordance to the specifications. [4] Software testing is a series of process which is designed to make sure that the computer code does what it was designed to do. Software testing is a destructive process. The main purpose of testing is quality assurance, reliability estimation, validation or verification. The other objectives or software testing includes. [3]

- The better it works the more efficiently it can be tested. [4]
- Better the software can be controlled more the testing can be automated and optimized.
- The fewer the changes, the fewer the disruption to testing. [5]
- A successful test is the one that uncovers an undiscovered error.
- Testing is a process to identify the correctness and completeness of the software. [5]
- The general objective of software testing is to affirm the quality of software system by systematically exercising the software in carefully controlled circumstances. [4]

There is big need of Software testing as described, for example while making food, it’s ok to have something extra, people might accept and eat the things we made and may also appreciate the work. But this isn't true for software project development. [7][8] If we fail to deliver a reliable, good and error free software solution, we fail in our project and probably we may lose clients. So in order to make surety of proper software solution, we go for testing. We test for any problem or error in the system, which can make software unusable by the client. [7] We make software testers test the system and help in finding out the bugs in the system to fix them on time.

Software testing is a process of measuring the quality of software developed. It is also a process of uncovering bugs in a program and makes it feasible. It is useful process of executing program. The diagram below represents some of the most prevalent techniques of software testing which are classified by purpose. [6]

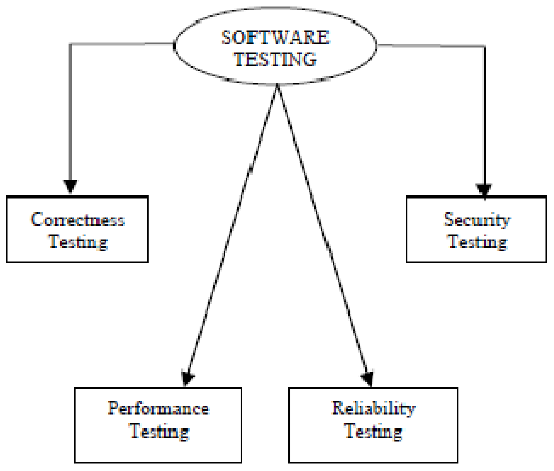


Fig.1 Represents different software testing techniques which are classified by purpose

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 2, February 2014

In order to ensure the software quality, one conducts software testing in every phase of the software development process. A complete software testing should cover the entire life cycle of software product. [1]

II. YIN-YANG TESTING THEORY

According to the definition of Yin-yang theory, test methods with still, obscure and organic characteristics are called Yin testing, for example, static testing, black box testing and performance testing; those which are dynamic, obvious and functional test methods are Yang testing, for example, dynamic testing, white box testing and function testing. This classification is shown in table 1.

According to Yin yang theory, testing is a mixed approach in which the testing of software various processes got mixed and affects heavily the other testing processes.

TABLE. I
SOFTWARE DEVELOPING AND SOFTWARE TESTING PHASE CORRESPONDENCE COMPARISON [1]

No.	Software developing phase	Corresponding software testing phase
1	User requirements	User Requirements verification and confirmation Acceptance test design
2	Requirement analysis and system design	Requirements verification and confirmation System test design
3	Conceptual design	Conceptual design verification and confirmation Integration test design
4	Detailed design	Detailed design and verification Unit test design
5	Coding	Unit test
6	Module integration	Integration test
7	System structure implementation	System test

In any software testing process, the test cases are either with Yin test or Yang test properties; this is the inherence theory of Yin-Yang software testing.

TABLE. II
YIN- YANG THEORY FOR SOFTWARE TESTING

Yin test	Static testing	Does not need to run the program, it's still.
	Black-box testing	Does not care about the internal logic of the program, it's obscure
	Performance testing	Cares about time performance, space performance and stability, with organic properties
Yang test	Dynamic testing	Do run the program and its dynamic
	White-box testing	Study internal logic and process procedure
	Function testing	Checks whether the software matches customer requirement

In development phase, engineers often perform code review, this is static testing(Yin testing) and also white box testing(Yang testing), it contains both properties of Yin and Yang testing; During the phase of unit test and integration test, test engineers usually perform white box testing, which read through parameters and structural to check validity (Yang testing), at the same time, they also need to conduct black box testing to verify the proper functionality of the modules (Yin testing), it contains both properties of Yin and Yang testing; when carrying out system testing and acceptance testing, engineers not only test function (Yang testing), but also test performance(Yin testing), it contains both properties of Yin and Yang testing. The inherence theory of Yin-Yang software testing explains that in the software testing process contains both Yin and Yang properties, that is to say, must conduct Yin test and Yang test, satisfying the inherence of testing. [1].In our research we are enhancing the similar theory approach to find software testing behaviour with different tests.

III. PROPOSED WORK

In our Research we will focus on improving this model description by adding more user end experience in acceptance testing. Various testing will be fetched on industrial experience based on Java applications. Various experiences from industrial companies will be fetched and will be implemented according to the proposed optimized Yin-Yan Theory for software testing. We will also try to introduce more testing dependencies. In User testing, we will add offline and online dependencies test which will be helpful in finding issues in overall acceptance testing phase. Software testing will be done by various tools and will link to proposed theory. Test model selection and test volume evaluation method will be applied to the software testing work of Industrial applications and will compared with traditional method.

IV. METHODOLOGY

In this paper, refined approach with Yin- Yan theory will be shaped with adding dependencies of online and offline test experiences in user testing phase to improve acceptance testing will be come into act. We will find the bugs in software by implementation of various tests described in Yin- Yan Theory and distribution table will be created to fetch accurate results in testing phase. Software testing packages will be fetched form industrial experiences. Preferably our testing will be based on small scale industry so that we can have local resources for any experimentation. For fetching various testing experiences, we will approach some known companies based on testing business in Small scale industry. Validation will be based on the number of bugs found and workload compared to traditional testing.
In first step of the development and testing, we have focused on the testing techniques as per given below.

IV.I. Testing strategies used for Research

Extensive study survey did regarding various types of testing and understands different types of testing concepts.

Black Box Testing- In this software testing techniques in which functionality of the software under test (SUT) is tested without looking at the internal code structure.

White Box Testing- This is testing of a software solutions internal coding and infrastructure. It focuses primarily on strengthening security, the flow of inputs and outputs through the application, and improving design and usability.

Static Testing: Here manually checks the code, requirement documents, and design documents to find errors.
Dynamic Testing: It checks for functional behaviour of software system and executes the software and validates the output with the expected outcome.

Functional Testing: Functional testing verifies that each function of the software application operates in conformance with the requirement specification. This testing mainly involves black box testing and it is not concerned about the source code of the application.

Acceptance Testing: User acceptance is a type of testing performed by the Client to certify the system with respect to the requirements that was agreed upon. This testing happens in the final phase of testing before moving the software application to Market or Production environment.

Security Testing: Security Testing ensures that system and applications in an organization are free from any loopholes that may cause a big loss. Security testing of any system is about finding all possible loopholes and weaknesses of the system which might result into loss of information at the hands of the employees or outsiders of the Organization.

Unit Testing:-Unit testing of software applications is done during the development (coding) of an application. The objective of unit testing is to isolate a section of code and verify its correctness.

Integration Testing: In integration testing, individual software modules are integrated logically and tested as a group.

Performance Testing: Here we did load testing to test our application performance. Load of 200 users to saw behaviour of application.

IV.II.APACHE JMETER

APACHE JMETER is a tool used for performance testing. The Apache JMeter™ is open source software, a 100% pure Java application designed to load test functional behaviour and measure performance. Apache JMeter may be used to test performance both on static and dynamic resources (Files, Web dynamic languages - PHP, Java, ASP.NET, etc. - Java Objects, Data Bases and Queries, FTP Servers and more). It is used to simulate heavy load on a server, group of servers, network or object to test the strength or to analyse overall performance under different load types. We can use it for graphical analysis of performance or to test our server/script/object behaviour under heavy concurrent load. Hence it is an Apache project that can be used as a load testing tool for analysing and measuring the performance of a variety of services, with a focus on web applications.

Apache JMeter features include:

- ➢ Ability to load and performance test different server and protocol types:
  - Web - HTTP, HTTPS
  - SOAP
  - FTP
  - Database via JDBC
  - LDAP
  - Message-oriented middleware (MOM) via JMS
  - Mail - SMTP(S), POP3(S) and IMAP(S)
  - Native commands or shell scripts
  - TCP
- ➢ Complete portability and 100% Java purity.
- ➢ Full multithreading framework allows concurrent sampling by many threads and simultaneous sampling of different functions by separate thread groups.
- ➢ Careful GUI design allows faster Test Plan building and debugging.
- ➢ Caching and offline analysis/replaying of test results.
- ➢ Highly Extensible core:
  - Pluggable Samplers allow unlimited testing capabilities.
  - Several load statistics may be chosen with pluggable timers.
  - Data analysis and visualization plug-in allow great extensibility as well as personalization.
  - Functions can be used to provide dynamic input to a test or provide data manipulation.
- ➢ Scriptable Samplers (BeanShell, BSF-compatible languages and JSR223-compatible languages).

V. EXPERIMENTAL RESULTS

Our initial work starts with selecting small scale companies for fetching various experiences and projects from industry based on software testing. We fetched some good projects from industry and we will use fetched software codes for further testing according to our proposed work.

Initial fetched project is known as secure communication which contains secure communication of clients with server in an encrypted manner. This project code is written in java.

This project contains basic structure with following conditions:

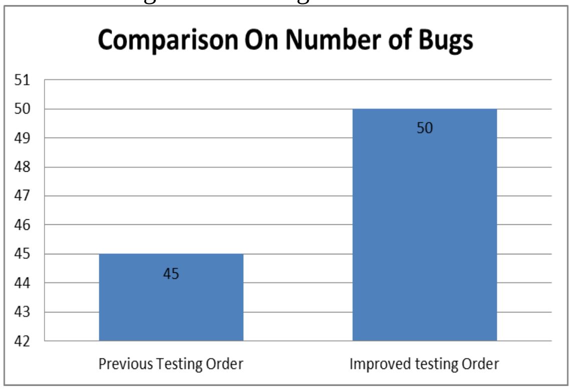
1. A Secure communication is the requirement which can be useful in providing better communication then already existing algorithms.
2. Complexity should be less.
3. Should be fast enough to process 100 clients.

Second project fetched is the based on testing Aspect oriented coding in cyvis tool. Again it is based on java.
This project contains basic structure with following conditions:

1. Development of the Aspect software is required with Aspect J as base Language. Minor alterations and suggestions by developers are acceptable.
2. Development of similar software is required in Object Oriented with Java Language as base.
3. Vision is to refine previous version that is in Java. Complexity is the concern and should be less with aspect changes.

4. Should support at least 10 client machines with core 2 duo processors installed.
5. Management will provide timely guidelines through official mail.
6. No other language is acceptable as company core platform software based on java.
7. Reflection of class difference reports will be required with analysis under cyvis tool.
8. Any suggestion from developers regarding testing is open.
9. In our further step we will implement various testing techniques to test these projects

Such test model method had been applied to the software testing work of 'Web Drive Explorer' application. Here is the comparison between previous testing model and optimized testing model previously 45 bugs were found and in our testing model 50 bugs are found.



Hence proved that our testing model is better than previous testing model. Below figure shows the difference of number of bugs found in previous and proposed scheme for software testing

VI. CONCLUSION

In our continuous research we are working on software testing the fetched projects according to Yin-Yang theory and will follow the test require and need to be implemented. Regression testing will be prime focus and will try to reduce as many faults possible and as many bugs we can detect. To make software testing effective, test cases are the good option which can control the resources used in testing and can provide good level of quality. In this research we have used offline and online applications built in JAVA platform for providing the optimized Yin-Yang theory for software testing. We have added online and offline application options to traditional Yin-Yang theory. Such test model selection and test volume evaluation method had been applied to the software testing work of Java application, and compared with traditional Yin-Yang testing mechanism.

The Yin-Yang testing used 202 test cases, discovering 45 bugs, whereas proposed method discovers 50 bugs out of 23 test cases done in the testing phase. The resultant test provide better level of software testing by increases the efficiency of the testing by 40 % as compared to traditional Yin-Yang software testing. From the experimental test effect and workload, the Yin-Yang test method is rational but the proposed optimized Yin-Yang method is more effective, especially when testing small scale applications.

VII. FUTURE SCOPE

In this research we have followed the traditional Yin-Yang Theory for software testing with some more features addition. The online and offline testing provide good customer feedback and provide better acceptance testing by customers. We have not considered the multiple applications for the various tests based on the multiple dependencies of hierarchy. Future work could be providing the better way to process Yin-Yang for gaming application which have multiple dependencies so that we can check the performance of fast changing and dynamic applications

REFERENCES

[1] Fangchun Jiang, Yunnan Lu, “Software testing model selection research based on Yin-Yang testing theory”, International Conference on Computer Science and Information Processing (CSIP), IEEE, Vol.9, 2012.
[2] Mohd. Ehmer Khan, "Different Forms of Software Testing Techniques for Finding Errors”, IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 3, No 1, May 2010.
[3] Introduction to software testing available at http://www.onestoptetsing.com/introduction/
[4] Software testing techniques available at http://pesona.mmu.edu.my/~wruslan/SE3/Readings/G B1/pdf/ch14-GB1
[5] Paper by Lu Luo available at http://www.cs.cmu.edu/~luluo/Courses/17939Report. pdf
[6] Software testing by Jiantao Pan available at http://www.ece.cmu.edu/~roopman/des- 899/sw_testing/
[7] Sahil Batra, Dr. Rahul Rishi, “Improving Quality Using Testing Strategies”, Journal of Global Research in Computer Science, Volume 2, No. 6, June 2011.
[8] Cem Karner, “Testing Computer Software”, 1993.
[9] Sheetal Thakare, Savita Chavan, Prof. P. M. Chawan, “Software Testing Strategies and Techniques”, International Journal of Emerging Technology and Advanced Engineering, pp. 567-569, Vol. 2, Issue. 4, April 2012.
[10] Abhijit A. Sawant, Pranit H. Bari and P. M. Chawan, “Software Testing Techniques and Strategies”, International Journal of Engineering Research and Applications (IJERA), pp. 980-986, Vol. 2, Issue 3, May-Jun 2012.