An Innovative Approach to Investigate Various Software Testing Techniques and Strategies

Syed Roohullah Jan[1], Syed Tauhid Ullah Shah[1], Zia Ullah Johar[1], Yasin Shah[1], Fazlullah Khan[1]

[1]Department of computer sciecne, Abdul wali university Mardan, KPK, Pakistan

ABSTRACT

Software testing is a way of finding errors from the system. It helps us to identify and debug mistakes, errors, faults and failures of a system. There are many techniques and strategies emerged since the concept of software development emerged. The aim of testing is to make the quality of software as efficient as possible.in this paper we discuss most widely used techniques and strategies. Where they can be used and how they can be used. How they work and how they differ (from each other).They are the following. Techniques: Black Box Testing, White Box Testing, And Grey Box Testing. Strategies: Unit Testing, System Testing, And Acceptance Testing.

Keywords : software testing, testing techniques, testing strategies, black box testing, white box testing, gray box testing, unit testing, integration testing, system testing.

I. INTRODUCTION

Software testing is a verification process for the assessment of software quality and a process for achieving that quality (Naik & Tripathy, 2013). An important activity of software development. It is old as the development of the software.

To determine the quality of software it depends upon how the software system is tested. Organizations and testers suggest giving 40-50% of their resources (time and budget) on testing. To achieve high level of reliability, maintainability, availability, security, Survivability, portability, capability, efficiency and integrity the system must need to be properly tested. Software testing give us an absolute guarantee that the system work as specified. Critical and modern software system must need to be correct and provide as much functionality as specified. Testing is in the vast interest of developer, tester, organization and end user. There are many software testing techniques and strategies that are used for testing. Which we discuss in our paper.

II. METHODS AND MATERIAL

1. Objective of software testing

A. Detection

Through detection all the insufficiencies, errors, flaws and failures present in the system are identified. Detection define the capableness and limitations applied on its functionality, quality of every component the system have, the products which works and the overall system. For the detection of errors, faults and failures several times testing is to be performed on the system with wrong input to ensure what will happened and what should not.

B. Prevention

An information providing us details about preventing and to reduce the errors and number flaws for the purpose to Elucidate overall specification and required performance of system. It defines us the causes because of which error occurs and preventing them from occurring in future.

IJSRSET1622210 | Received : 31 March 2016 | Accepted : 09 April 2016 | March-April 2016 [(2)2: 682-689]

C. Quality improvement:

“Quality cannot be achieved by assessing an already completed product”. [1] In critical environments the unreliability and poor quality of a software system can kill and can cause disasters. If the system is not bug free. Effective test ensure improvement in the quality of software and help to minimize ratio of errors.

D. Verification:

To ensure that the software provides the required functionality. Verification is done when the development process starts. Verification main goal is to meet the requirements which is specified. Verification basically questioned, Are we building the product right?

E. Validation:

Validation ensure that we are evaluating the required software as specified by user. Validation can be done during the development process or when development become complete. Validation questioned Are we building the right product?

2. Literature Review

Software testing a much activity of software development life cycle. It ensure us that the system will provide the required functionality. For this purpose many different software testing techniques and strategies are used. These techniques are White box, Black box and Grey Box testing. Using Black box the tester test the requirements and final result. Using White box the tester test source code/internal design. While testing with gray box, it combines the advantages of both white and black box testing.

Unit testing is testing that test only small units. Different integrated modules are tested using integration testing. While to test the system as a whole is done by System testing.

In this paper we compare and discuss these techniques and strategies that which one is to be used in which condition. We discuss them in great details.

3. Software testing techniques.

A. Black box testing:

A software testing technique which play an important role in software testing. While performing black box testing, tester don’t have any knowledge of internal design and no access to source code. Tester have only knowledge of system architecture. This technique is to be used to ensure that all inputs needed by the system is accepted in specified manner and provide correct output. [2]

Most widely known black box software testing techniques are the following.

1) Boundary value analysis:
   There is a possibility that the system may be fail on boundary. Because there are chances of error that the programmer done at the boundaries of equivalence classes. That why this techniques focuses on edges or values that are chosen at extreme boundaries.

2) Equivalence partitioning:
   This techniques help us to reduce the number of test cases. This techniques basically works on dividing the program input domain based on input values into equivalence classes. Test case are generated from these equivalence classes which are derived from the input domain.

3) Orthogonal Array Testing:
   Orthogonal array testing is a statistical way of testing.it is used where the input domain is very small and helps to reduce the number of test combinations. Variable is represented by columns and test cases is represented by rows.
4) Fuzzing:
   Fuzzing is a black box testing technique that is developed by Barton Miller in 1989 at university of Wisconsin. This technique is based on feeding random input to application. Fuzz testing technique can be used to find implementation bugs.in automated or semi-automated session fuzz test use malformed/semi-malformed data injection.
5) Graph based testing:
   Black box testing technique which starts by creating a graph. The graph is created from input modules.an

identifier is given to input modules. Through graph a connection is established between effect and its causes.

6) All Pair Testing:
   A sort of black box technique in which the purpose of test cases is to execute all discrete combinations which are possible for input parameter of each pair.to cover all the pairs we need to use a number of test cases.
7) State Transition Diagrams (or) State Graphs:
   A brilliant tool which is used to capture several types of system requirements and documented internal system design. This tool is also used to test state machine and to navigate GUI (graphical user interface).

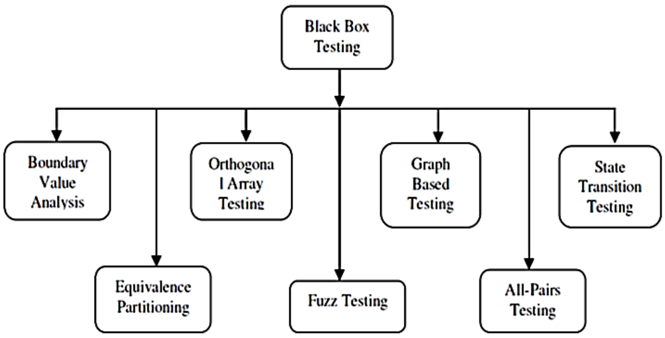

Figure 1. Different forms of Black Box Testing [3]

Advantages:

- By achieving reasonable testing number of test cases can be reduced.
- Testing is totally based on user's point of view.
- Tester and programmer are independent from each other.
- For large code segments it is the most efficient and well suited technique.
- No code accesses are required.

Disadvantage:

- Not well suited and efficient test.
- There are chances of duplication of tests that are already tested by programmer.
- Test cases are difficult to design without clear requirements.
- Some parts that are at back end are not tested at all.

B. White box testing:

This technique is also known as Structural Testing or Code-Based Testing [6, 7, 8, 9, 10, 11, 12]. The main focus of this testing technique is on investigation the internal logic and structure of source code.it can be applicable at integration, unit and integration levels. Using this technique it is most important and necessary for a tester to have complete knowledge of source code [13, 14, 15, 16].

Most widely known white box software testing techniques are the following.

1) Desk checking:
   Desk checking is a manual way to check the logic of programs. This test is usually done by the programmer using pen and paper to records the result of the test.
2) Code Walkthrough:
   It is a form of peer review in which review process is leaded by programmer and to spot errors other team member's questions about the system.
3) Formal inspection:
   To find errors in code and design this formal, economical and efficient technique is used. Led by third party contractor. This technique is the most formal one. Its main goal is to detect faults, side effects and violations.
4) Control Flow Testing:
   A fundamental and effective technique for all software. This is applied to almost every software.it is based on the structure of code coverage that is referred to the amount of program which has been tested. The criteria/target of this technique is 100% coverage.
5) Basis Path Testing:
   To evaluate logical complexity [17, 18, 19] of procedural design this technique will be used. For testing each path of code which are independent is taken for testing. Flow graphs will be used to represent control flow in program.
6) Data Flow testing [20, 21, 22] :
   To know how about the program variables are defined and used this technique use the control flow graph.
7) Loop Testing:
   In loops error mostly occur nearly at the beginning and end. This technique focuses on the validity of loop construct (is there any possibility that loop successfully ends).

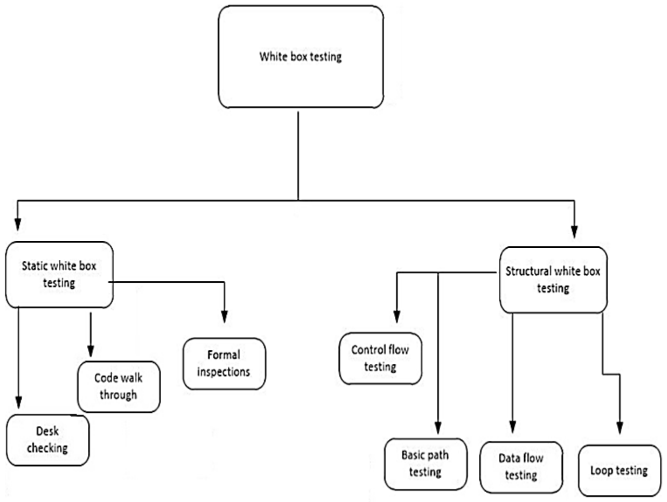

Figure 2. Different forms of White Box Testing [4]

Advantage:

- Best technique for code optimization.
- Execution must be required for every loop and their body.
- Hidden code errors are exposed by removing extra lines of code.
- To maintain validity internal data structure are be exercised.

Disadvantage:

- Costly because a fully skilled tester is needed.
- The cases which is omitted in code are missed out.
- It is very difficult to find out the hidden errors by looking at every bit of code. This may create problems and result in failure of software.

C. Gray box testing:

This is testing technique in which there is little knowledge of internal work of application. This technique is language and platform independent. Grey box combines the benefits of white box and black box testing. To design test cases it use algorithms and internal data structure less than white box testing but more than black box testing.

Grey box=white box + black box, To test a piece of software against it specification but having some knowledge of internal working this technique will be used. Widely used in integration testing but it can also applied to most testing phases.

Most widely known grey box software testing techniques are the following.

1) Orthogonal Array:

This testing technique use as subset of all possible combinations. This is systematic and statistical.

2) Matrix Testing:
In this technique the project static report is stated.

3) Regression Testing:
Regression testing implies running of test cases. If new changes are to be made in the system.

4) Pattern Testing:
To verify that the application is good for its design and architecture. This technique will be used.

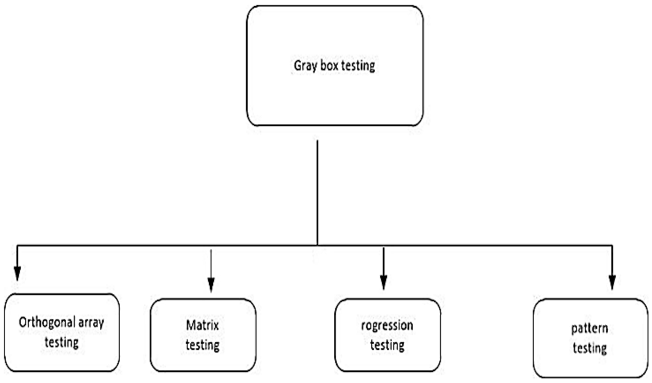

Figure 3. Different forms of Grey Box Testing [5]

Advantages:

- This testing technique totally done test from user point of view rather than from designer point of view.
- Efficient test scenarios are design by tester.
- Whenever possible gives both the benefits of white-box and black-box testing.
- Reasons about implementation are carefully given by developer.

Disadvantages:

- Paths of many programs remain untested when using this technique.
- If a test is already done by designer then the test become redundant.
- Because of the unavailability to access source code cause limit the test coverage.

Table 1 : Comparison between three forms of testing techniques

S No.	**White box testing**	**Black box testing**	**Gray box testing**
1.	This technique is suited for algorithm.	This technique is not suited for algorithm.	This technique is not suited for algorithm testing.
2.	Tester must have full knowledge of internal working.	Tester only need know about analysis fundamental aspects i.e. no knowledge of internal work required.	Tester only need partial knowledge of internal working.
3.	In this technique granularity is high.	In this technique granularity is low.	In this technique granularity is medium.
4.	This technique is also known as Code-based testing, Clear-box testing and structural testing.	This technique is also known as Closed-box testing and data driven testing.	This technique is also known as translucent testing.
5.	This testing technique is performed by testers and developers.	This testing technique is performed by testers and developers and end user.	This testing technique is performed by testers and developers and end user.
6.	This testing technique is most time consuming and exhaustive.	This testing technique is least time consuming and exhaustive.	This testing technique is somewhere between two.

III. RESULTS AND DISCUSSION

Software Testing Strategies

A. Unit Testing

The smallest testable part (i.e. procedures, interfaces, functions and classes) of an application is known as unit. This unit are tested to determine are they are ready to use? System and integration level are its complements.it can be done mainly as well as automatically. Software developer write and execute unit test to make sure that it behaves as expected in design and specification.to identify and fix errors test cases for all methods and functions are written. Unit testing is done by developers.

Different integration testing techniques are the following.

1) Module Interface test:
   To check out that information is flowing in proper order in to the program unit and out in proper order or not this method is used.
2) Local data structures:
   This is used to check if the data which is stored temporarily are stored proper and maintain integrity or not.
3) Boundary conditions:
   To check the module boundary condition in order to make them to run program properly at boundaries.
4) Independent paths:
   To ensure that all statements in a module are executed at least once all independent path are to be tested.
5) Error handling paths:

   To ensure that errors are properly handled error handling paths are to be applied after successful completion of various tests.

Advantages:

- This testing is performed by application developer.
- It is very cost effective.
- To achieve high level structure of code.
- It makes it easy to test part of system without waiting for the availability of other parts.
- Less cost required to detect and remove error.
- Because of testing small units of code area it is simple technique.

Disadvantage:

- Using unit testing it is difficult to catch every bug in the program. Because unit are tested separately.
- It is time consuming.
- Using this system and integration errors may be missed.
- Writing good test cases are too difficult.
- Because of mistakes done by developer effect the overall system.

B. Integration Testing

To merge software module and to test them as group is known as integration testing. This testing is between system and unit testing.

This technique uncover errors as well as construct program structure at same time. The goal of this technique is to test the interface between modules/units.

Different integration testing techniques are the following.

1) Top down integration:
   To construct program structure this incremental approach is used. Start from top most module and progress through lower modules.to integrate and test all the modules the process repeat itself.
2) Bottom up integration:
   Start testing from the innermost unit of application and moves up.to test the overall application as single unit integration continues until all modules are integrated.
3) Big bang testing:
   This is a non-incremental approach. Start with combining all components and test overall program and a set of errors as a whole. This technique is infinite because errors occur after correction.

Disadvantage:

- Difficult to debug errors.
- User have no access to application until late in development cycle.
- Much throwaway coding are required.

Advantage:

- Testing is based on previously tested modules/units.
- Modules are separately tested.
- Testing is done by integrating previously tested modules.

C. System Testing:

This type of testing is performed when system become complete with its specified requirements. This testing doesn't required knowledge about source code. This is a series of different tests. Purpose of this kind of testing is to exercise fully the application. Each test which is to be performed on the system have different purpose. This verify that system are properly integrated and able perform required functionality.

Different system testing techniques are the following

1) Recovery testing:
   This is a type of non-functional testing. The main purpose is to force the software to make it fail to verify how much fast and how much efficiently recovery is performed. If recovery is automated for correctness of the system check pointing mechanisms, re-initialization, restart and data recovery are evaluated.

2) Security Testing:
   It ensure that there is no loopholes that cause a big loss. This test to find all the loop holes and weakness that the system have. Tester attack the system with customer software that are design to breakdown the defense that the system have.

3) Graphical user interface testing:
   Using this technique testing is performed on user Graphical user interface to ensure that it meets as specified. This involves testing screen controls like icons, menus, tool bars, buttons, menu bar, windows and dialog boxes etc.
4) Compatibility testing:
   This is a non-functional testing. That is done to check the software product Compatibility by running it on different operating systems, networks, hardware, browsers and databases.

Advantages:

- System is tested as a whole.
- To ensure that its meets the technical and functional specification.
- Test is done in the environment that is close to the environment where software is deployed.
- Test validates and verifies application architecture and business requirements.

Disadvantages:

- No knowledge of source code and system architecture.
- Test don't have any knowledge of how system is integrated.
- Time consuming.

Table 2 : Comparison between three forms of testing techniques

	Unit testing	**Integration testing**	**System testing**
1	Doesn't occur after or before anything.	Occur after unit and before system testing.	Occur after integration testing.
2	Pay attention only to the behavior of single independent modules.	Pay attention to integration among different module.	Pay attention to overall system functionality.
3	Complex scaffolding means frame are required.	Some scaffolding means frame are required.	Doesn't scaffolding means frame are required.
4	Kind of white box testing	Kind of both white and black box testing.	Kind of black box testing.
5	Test visibility of code in detail	Test visibility of integration structure.	Doesn't testing visibility of code.
6	Start from module specification.	Start from interface specification.	Start from user requirement specification.

IV. CONCLUSION

This paper describes what is testing and what are the techniques and strategies for software testing. Software testing is an activity that executes software system to identify errors and determine that it meets requirements. Provide a view of the system that are independent. Risk which can be in the implementation of software it allow the business not only to appreciate but also provide understanding about risk. Black testing technique is to be used where testing is based on user requirements and for large code segments. White box testing is used to find implementation and internal code errors in order to remove extra lines of unwanted code. Grey box testing is used to test interface and functional specification. Unit testing is used for the software and hardware units that are individual or related units that are grouped together. Integration testing are to be used to test either software either hardware components or combination of them for evaluation of interaction between them. System testing is to be used to test the system as a whole. We further explain and compare different techniques and strategies. Such as Black-Box testing, White-Box testing and Grey-Box testing. We also compare Unit testing, Integration testing and System testing.

V. REFERENCES

[1] William E.Lewis, W.H.C.Bassetti "Software Testing and Continuous Quality Improvement"2nd Edition, Auerbach Publications, 2004

[2] R.S. Pressman & teal, "Software Engineering A Practitioner's Approach" 6/e; Chapter 14: Software Testing Techniques, 2005

[3] Mohd. Ehmer Khan, "Different Approaches to Black Box Testing Technique for Finding Errors," IJSEA, Vol. 2, No. 4, pp 31-40, October 2011

[4] Nidhra, Srinivas, and Jagruthi Dondeti. "Black box and white box testing techniques- A Literature." International Journal of Embedded Systems & Applications 2.2 (2012).

[5] Grey Box Testing from Wikipedia available at http://en.wikipedia.org/wiki/Gray_box_testing

[6] M. A. Jan, P. Nanda, X. He and R. P. Liu. 2013. "Enhancing lifetime and quality of data in cluster-based hierarchical routing protocol for wireless sensor network", 2013 IEEE International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing (HPCC & EUC), pp. 1400-1407.

[7] M. A. Jan, P. Nanda, and X. He. 2013. "Energy Evaluation Model for an Improved Centralized Clustering Hierarchical Algorithm in WSN", in Wired/Wireless Internet Communication, Lecture Notes in Computer Science, pp. 154–167, Springer, Berlin, Germany.

[8] F. Khan, K. Nakagawa. 2012. "Performance Improvement in Cognitive Radio Sensor Networks" in the IEICE Japan.

[9] M. A. Jan, P. Nanda, X. He and R. P. Liu. 2014. "PASCCC: Priority-based application-specific congestion control clustering protocol," Computer Networks, vol. 74, pp. 92-102.

[10] Mian Ahmad Jan and Muhammad Khan. 2013. A Survey of Cluster-based Hierarchical Routing Protocols, IRACST–International Journal of Computer Networks and Wireless Communications (IJCNWC), Vol.3, pp.138-143.

[11] F. Khan, K. Nakagawa. 2013. “Comparative Study of Spectrum Sensing Techniques in Cognitive Radio Networks” in World Congress on Computer and Information Technology, pp.1-8

[12] Mian Ahmad Jan and Muhammad Khan. 2013. Denial of Service Attacks and Their Countermeasures in WSN, IRACST–International Journal of Computer Networks and Wireless Communications (IJCNWC), vol. 3, April 2013.

[13] M. A. Jan, P. Nanda, X. He and R. P. Liu. 2015. “A Sybil Attack Detection Scheme for a Centralized Clustering-based Hierarchical Network,” in Trustcom/BigDataSE/ISPA, Vol.1, PP-318-325, IEEE.

[14] M. A. Jan, P. Nanda, X. He, Z. Tan and R. P. Liu. 2014. “A robust authentication scheme for observing resources in the internet of things environment” in 13th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 205-211, IEEE

[15] F. Khan, S.A. Kamal, F. Arif, “Fairness Improvement in long-chain Multi-hop Wireless Ad hoc Networks” in IEEE ICCVE 2013, Las Vegas, USA 2-6 December, 2013

[16] Ali, H., Saeed, A., Jan, S.R.U., Khan, A.U., & Khawaja, A. 2012. “Internet Connectivity using Vehicular Ad-Hoc Networks”

[17] Ali, R., Ali, H., Salman & Iqbal, S. 2014. “A Novel Survey on: Mobility Based Routing in Vehicular Ad-Hoc Networks (Vanets)”, Journal of Applied Environmental and Biological Sciences, pp. 487.

[18] Mian Ahmad Jan, “Energy-efficient routing and secure communication in wireless sensor networks” PhD Thesis, 2016. http://hdl.handle.net/10453/43497

[19] F.Khan 2012. “Secure communication and routing architecture in wireless sensor networks” in IEEE 3rd Global Conference Consumer Electronics (GCCE), pp 647-650

[20] K.Nakagawa F. Khan, F. Bashir. 2012. “Dual Head Clustering Scheme in Wireless Sensor Networks” International Conference on Emerging Technologies (ICET), pp. 1-5.

[21] M. A. Jan, P. Nanda, X. He and R. P. Liu. 2016. A Lightweight Mutual Authentication Scheme for IoT Objects, IEEE Transactions on Dependable and Secure Computing (TDSC), “Submitted”.

[22] M. A. Jan, P. Nanda, X. He and R. P. Liu. 2016. A Sybil Attack Detection Scheme for a Forest Wildfire Monitoring Application, Future Generation Computer Systems (FGCS), “Submitted”.