ISSN:0975-9646

Software Testing

Uma Nath Yadav, Ashutosh Rai,,Priyaranjan Verma

B.Tech(IT)
ITM Gorakhpur

Abstract:Software testing is the process of validating and verifying that a software program/application meets that requirement that guide its design and development, work as expected, and can be implemented with same characteristics. Many attempts to obtain software testing based on methods from the field of design of experiments have been recommend as a mean of providing high coverage at relatively low cost. Many tools to generate aim pairs, or higher n-degree combination, of input values have been developed and demonstrated in a some applications, but little empirical evidence is available to help developer in evaluating the effectiveness of these tools for particular complexities.

1. INTRODUCTION:

Software testing is a process of finding errors or bug in the software/program and it involves any activity & evaluating an attribute or capabilities of a program/system or software and determining that it meets its required results.

The software testing is discovering errors/bugs before the user does a good tester is one who is passed in making the system fail. The aim of a tester is opposite of the authors/developer. Basically software testing is consists the following equation-

Software Testing = Software verification + software validation.

1.1. SOFTWARE VARIFICATION:

Software verification is a human test activity as it involves in the process of evaluating a system or component to determining whether the software / product of a given development phase satisfy the condition of starting phase or not.

1.2. SOFTWARE VALIDATION:

The software validation is a process of evaluating a system or its components are satisfied the specified requirements and involves executing the actual software.

2. GOAL OF SOFTWARE TESTING:

The main aimed of software testing at evaluating an attribute are capability of a program or system and demining that it meets its required results.

It stability a way of finding effective testing in order to deliver good quality of software and also consists the validation of the products.

3. LEVEL OF SOFTWARE TESTING:

The software testing level is divides into following categories-

a. Unit Testing
b. Integration Testing
c. System Testing
d. Regression Testing

3.a. UNIT TESTING:

The unit testing is the process of taking a single module from a large set of module of a software/product. through the unit testing a module of the software is tested individually so this testing becomes easier .

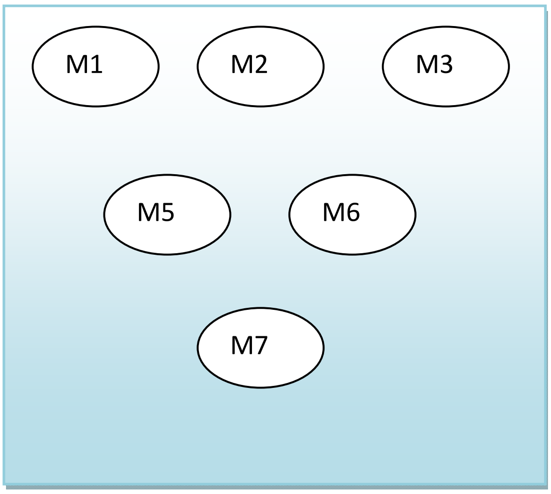

Fig. Unit Testing

The unit testing testing's is also known as white box testing. The unit testing is test the internal functionality of the software. It design a test tool for the testing o module before the implementing a software.

Importance of Unit Testing:

For any software, It is very highly ineffective and inefficient. It is more exhausted and interface errors are eliminated.

3.b. INTEGRATION TESTING:

Integration Testing is the testing of software components after they have been integrated and it test perform after the unit testing because “some bugs that can be discovered as the units are integrated are impossible to find when testing isolated units"

The integration testing are perform through following ways-

3.b.1. TOP DOWN INTEGRATION:

The top down integration is start with the root node of the tree or program/ software any lower level unit/node that is called by main program appears as stub where stubs are pieces of code/program/software.

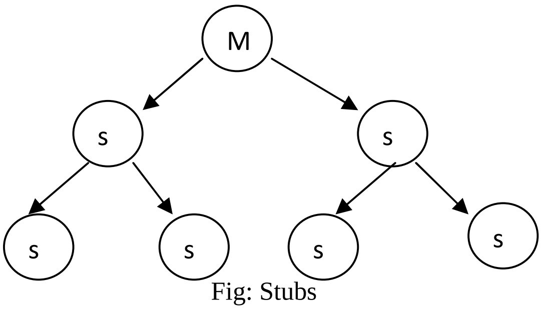

Fig: Stubs

Where M= main program
S= stubs

Then the stub required = (number of nodes-1).

ADWANTAGE OF TOP DOWN INTEGRATION:

The major advantage of this top-down method is the fact that a system prototype can be developed early on in the project process. This is a very attractive property of this integration testing technique as usually the client of a software company will have little or no software engineering knowledge

And therefore if the client cannot see something tangible then it would be easy for the client to assume that little or no work is being completed.

3.b.2. BOTTOM UP INTEGRATION:

The Bottom up integration starts with the leaves of the decomposition tree and test then with specially coded drivers. Any higher nodes that are directly connected to the nodes being tested are constructed as drivers.

Let if D = driver
M = modules

Then drivers required = (Number of nodes – number of leaves).

ADWANTAGE OF BOTTOM UP INTEGRATION:

The major advantage of this method of integration testing is that the program itself is fully functional at every stage. This is in contrast from the top-down method, where as a prototype can be created at an early stage, but it will have little or no functionality

3.b.3. BIG BANG INTEGRATION:

The big bang integration testing is consist all modules or builds are constructed and tested independently of each other and when they are finished, they are all put together at the same time.

ADWANTAGE OF BIG BANG INTEGRATION:

The main advantage of this approach of the big bang integration is that it is very quick as no drivers or stubs are needed.

3.b.4. THREADED INTEGRATION:

The threaded integration is an incremental testing technique that identifies major processing function that the product is to perform and maps these function to modules implementing then. Each processing function is called as “Thread”. A collection of related threads is called a “Build”. Build may be serving as a basis for test management.

3.c. SYSTEM TESTING:

The process of system testing is the integrated software and hardware both to verify that the whole system meets its specified requirement.

The system testing must be consisting the system capabilities rather than component capabilities. The system test should be developed and performed by a group independent of people who developed the code of software. It have repeatable in nature. The system test progress must be planned and tracked.

3.d. REGRESSION TESTING:

The regression testing is done during in maintenance. The process of executing previously define test cases on the modified program to assure that the software changes have not another affect the program’s previously existing function.

The regression testing includes the following type of error targets.

a) Data corruption errors
b) Inappropriate control sequencing errors
c) Resource contention.
d) Performance deficiencies.

The main purpose of regression testing is to increase confidence in correctness of the modified program and locate the errors in the modified program. Through the regression testing we are verify the modified software quality and reliability.

4. LIMITATION OF SOFTWARE TESTING:

The testing can so presence of errors not their absence and no any matter how hard you try, would never find the last bug in an application. There is more than one possible path through the program to test. Various testing techniques are complementary in nature and it is only through their combined use that one can hope to detect most errors.

5. CONCLUSION:

We have presented a comprehensive overview of software testing concepts, techniques and processes. This study compress the stream less of code reading structural testing in three aspect of software testing: fault detection effectiveness, fault detection cost, and classes of fault detect. the main aspect of the software testing is provide to help software tester and software developer in developing software without any error/fault/bug and through software testing we can improve the quality of software/product and customer satisfaction.

REFERENCES:

1. KANER, C; FALK, J; NGUYEN H. 1993 Testing Computer Software
2. JORGENSEN, P. 2002. Software Testing
3. Kieras, D.E. and Polson, P.G. An approach to the formal analysis of user complexity. International Journal of Man-Machine Studies, (1985)
4. Polson, P.G. A quantitative theory of human computer interaction. In J.M. Carroll (Ed.). Interfacing thought: Cognitive Aspects of Human-Computer Interaction. Bradford Books/MIT Press, Cambridge, MA, 1987.
5. TSAO, H; Wu, Y. 2003 Testing and Quality Assurance for Component-Based Software
6. BURNSTEIN, I. 2002. Practical Software Testing: A Process-Oriented Approach
7. MYERS, G. 2004. The Art of Software Testing
8. KAN, S. 2002. Metrics and Models in Software Quality Engineering

9. CRAIG, R; JACKIEL, C. 2002. Systematic Software Testing
10. http://www.wikipedia.com
11. http://www.softwaretestinggenius.com
12. http://www.4share.com