Significance of Software Testing Methods - Case Driven

Dr. B V Ramana Murthy[1], Dr. V Padmakar[2]

[1]Professor & Principal, Department of CSE, AAMEC, Hyderabad, India
[2]Associate Professor, Department of CSE , MIST, Hyderabad, India

Abstract—*Software Testing is a process of any software development which is used to measure the quality of developed software final product. It finds all errors, bugs and flaws in the developed software final product. In this paper we present software testing techniques for detecting the errors which are described by static testing and dynamic testing.*
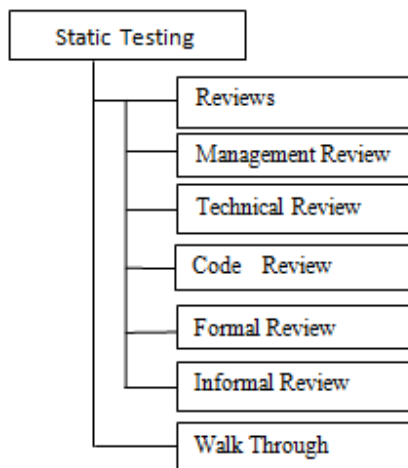
Keywords—Testing, static testing, dynamic testing.

I. INTRODUCTION

Software Testing Levels
1. Static Testing
2. Dynamic Testing

1. Static Testing:

It is the process of verifying or developing the right system or not. This static testing will be carried out with the help of reviews and walkthroughs**.** Verification[1] is to check whether software conforms to the specifications done by the development team at various development phases. During development phase the SRS document, Design document, Code document are reviewed to ensure that product is being developed using the process oriented approach.

- It is an in-house activity of the development organization.
- It is a quality assurance [2] activity which prevents the defects of the product.

i) **Reviews**

Examining a project related work or process related work is called review.

       Types of reviews:
       a) Management Review
       b) Technical Review
       c) Code Review
       d) Formal Review
       e) Informal Review

a) Management Review: This review will be conducted by top level or middle level management to monitor the project status. Those reviews are helpful for the management to take the necessary actions.

b) Technical Review: Technical Reviews will be conducted among the technical people to decide the best approach of implementation.

c) Code Review: This review will be conducted among the developers to decide the best approach of programming preparation.

d) Formal Review: If a review is carried out with a particular plan by following a systematic procedures and proper documentation then it is called as Formal Review.

e) Informal Review: If a review is conducted without following any procedures and documentation then it is called as Informal Review.

ii) **Walk Through**:

A step by step presentation conducted by the other or by the domain expert about a subject. KT (Knowledge Transfer) is the best example for walk through[1,2]

2. Dynamic Testing
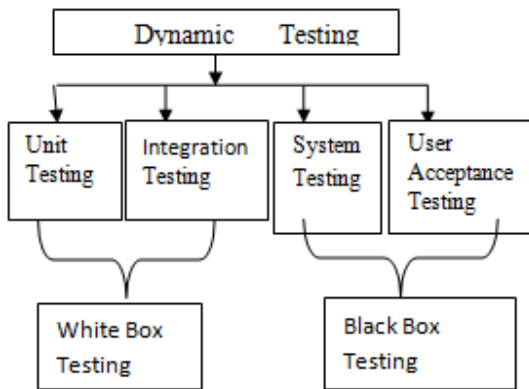
There are two types of methods
  i)   White box testing
  ii)   Black box testing

i)   White Box Testing
Testing[3] conducted by the developers on Coding to ensure the code coverage that the code is working as expected or not is called White box testing. Combination of unit & integration testing is called White box testing. White box[10] testing is also as Glass box, Clear box or Structural testing.

Unit Testing :The smallest testable portion in the source code of the application is called Unit Testing.
- Module Testing
- Component Testing (Functions, Procedures, Methods, Objects)

Integration Testing
Integration testing[4] is a software development process in which program units are combined and tested as groups in multiple ways. Once the unit testing is completed developers will integrate all source code units and check interaction among all units. This is called as Integration Testing.

**While conducting Integration testing follow the below subtests**
i) Top Down Approach
ii) Bottom Up Approach
iii) Hybrid Approach
iv) System Approach / Big Bang Approach

i)   Top Down Approach
Top down integration is primarily considering as an approach where modules are developed and of that modules always starting at the first level of the programming hierarchy and continuing towards the lower levels. Top

down is an incremental approach because we precede one level at a time.
Example: Login Page
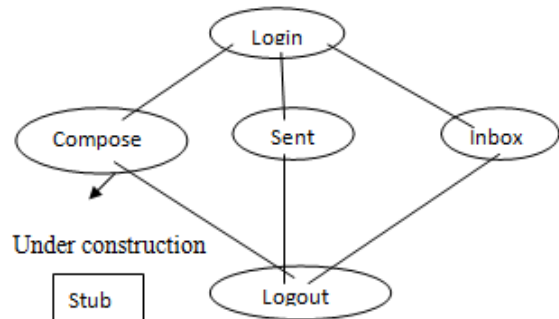
Under construction
Benefits
  1.   Having the framework, we can test major or supreme functions early in the development process.
  2.   Major benefit of this practice is that we include a partially working framework to show to the clients and to the top management too

Drawbacks
  1.   Impose stubs does not permit all the essential upward data flow.
  2.   The top level modules cannot be really tested perfectly and every time the stubs are replaced with the real modules, the modules which are calling should be properly re-tested again for integrity.

ii)   Bottom Up Approach
After unit testing of individual components the components are combined together into a system. Bottom-Up Integration: each component at lower hierarchy is tested individually; then the components that rely upon these are tested.
Component Driver: a routine that simulates a test call from parent component to child component
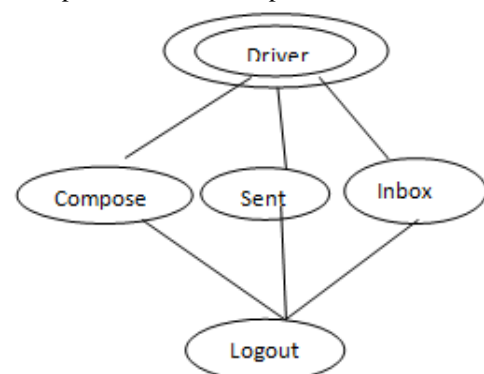
iii)   Hybrid Approach

Combination of top down and bottom up approach is called as Hybrid or Sandwich approach.

iv)   System Approach

System approach [1]is also known as Big Bang Approach. In Big Bang integration testing all components or modules are integrated simultaneously, after which everything is tested as a whole. In this approach individual modules are not integrated until and unless all the modules are ready. This approach is generally executed by the developers. In case any bug arises then the developers has to detach the integrated modules in order to find the actual cause of the bug.
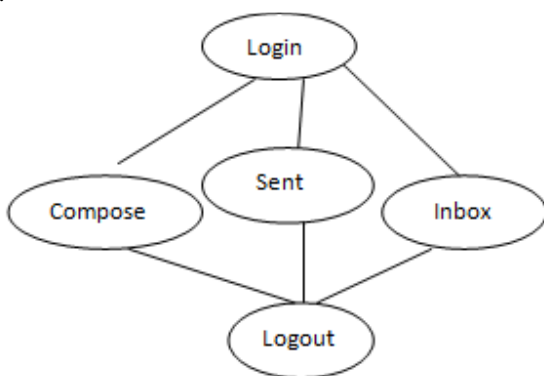
Note: From the above approaches we can follow any one of the approach based on the requirement.

Functional System Testing

In functional system[1,2] testing basically the testing of the functions of component or system is done. It refers to activities that verify a specific action or function of the code. It is a mandatory level testing.

During functional system testing, testers concentrate on below subtests:

i)   Sanity Testing
ii)   Real Testing
iii)   Retesting
iv)   Regression
v)   Database Testing

i)   Sanity Testing

Sanity Testing is the surface level testing where QA engineer verifies that all the menus, functions, commands available in the project and project are working fine. Sanity testing is carried out to check whether the bugs reported in previous build are fixed & there is regression introduced due to these fixes i.e. not breaking any previously working functionality[10]. The main aim of Sanity testing to check the planned functionality is working as expected. Sanity tests helps to avoid wasting time and cost involved in testing if the build is failed. After completion of regression testing the Sanity testing is started to check the defect fixes & changes done in the software application is not breaking the core functionality of the software. Typically this is done nearing end of SDLC i.e. while releasing the software. You can say that sanity testing is a subset of acceptance testing.

Here are the few consolidated points of Sanity testing:

- Sanity testing follows narrow and deep approach with detailed testing of some limited features.
- Sanity testing is typically non-scripted.
- Sanity testing is a sub-set of regression testing.
- Sanity testing is cursory testing to prove software application is working as mention in the specification documents & meets the user needs.
- Sanity testing is used to verify the requirements of end users are meeting or not.
- Sanity testing to check the after minor fixes the small section of code or functionality is working as expected & not breaking related functionality.

During this test, test engineers concentrate on basic functionalities of the Application like:

- Application basically working or not?
- Understandable or not?
- Consistent or not?
- Controllable or not?
- Simplicity

Example for Sanity Testing

There are five modules in a project like

- Login Page
- Home Page
- User details Page
- New User creation
- Task creation

So we have the bug in Login page like **User Name** field accepts less than 6 alpha numeric characters which are against the requirements. It is specified that username should not be below than 6 characters but as user name accepts less than 6 characters it is the **bug.**

- Now the bug is reported by the testing team to the developer team to fix it. When the developing team fixes the bug and passed it to testing team then the testing team checks the other modules of the application

ii)   Real Testing

Checking every functionality of the application is called Real Testing (checking size and type)

iii) Retesting

Retesting is executing a previously failed test against new software to check if the problem is resolved. After a defect has been fixed, retesting is performed to check the scenario under the same environmental conditions. During Retesting[3,4], testers look for granularity details at the changed area of functionality.

iv) Regression Testing

Regression testing is the process of testing changes to computer programs to make sure that the older programming still works with the new changes.

Example:

There are 3 modules in the project:

- Admin module
- Personal information
- Employment module

Suppose bug occurs in the Admin module like existing user is not able to login with valid login credentials (this is the bug).

- Now testing team sends the above mentioned bug to the development team to fix it and when development team fixes the Bug and hand over to testing team then testing team checks the fixed bug does not affect the remaining functionality of the other modules. So this is known as Regression testing.

v) Database Testing

Database testing is one of the major testing which requires tester to expertise in checking tables, writing queries and procedures. Testing can be performed in web application or desktop and database can be used in the application like SQL or Oracle. There are many projects like Banking, Finance, Health insurance which requires extensive database testing. Nowadays database is getting more complex due to the business logic which plays an important role for the applications. Testers should make sure that values have been added correctly after the implementation of the business rules. Database is the spine of the application and tester should make sure to test very carefully. It requires skill, proficiency and sound knowledge.

Example – Login and User Security

The validations of the Login and User security credentials need to take into consideration the   following things:

1. Whether the application prevents the user to proceed further in the application in case of a
- Invalid username but valid password
- Valid username but invalid password
- Invalid username and invalid password
- Valid username and a valid password

2. Whether the user is allowed to perform only those specific operations which are specified by the business requirements.
3. Whether the data secured from unauthorized access[7,8]
4. Whether there are different user roles created with different permissions.
5. Whether all the users have required levels of access on the specified Database as required by the business specifications.
6. Checking that sensitive data like passwords, credit cards numbers are encrypted and not stored as plain text in database.

Non Functional System Testing

After completion of functional system testing, test engineers concentrate on non functionalities of the application like **User Interface**, **Performance, and Compatibility, Security** etc.

i) User Interface Testing

User interface testing is a technique used to identify the presence of defects in a product/software under test by using Graphical User Interface (GUI)

Graphical User Interface (GUI) testing is the process of testing the system's GUI of the System under Test. GUI testing involves checking the screens with the controls like menus, buttons, icons, and all types of bars - tool bar, menu bar, dialog boxes and windows etc.

The following checklist will ensure detailed GUI Testing.

- Check all the GUI elements for size, position, width, length and acceptance of characters or numbers. For instance, you must be able to provide inputs to the input fields.
- Check you can execute the intended functionality of the application using the GUI
- Check Error Messages are displayed correctly
- Check for Clear demarcation of different sections on screen
- Check Font used in application is readable
- Check the alignment of the text is proper
- Check the Color of the font and warning messages is aesthetically pleasing
- Check that the images have good clarity
- Check that the images are properly aligned
- Check the positioning of GUI elements for different screen resolution.

Example of User Interface Testing for Mobile Application

User Interface Testing is to test the characteristics of a mobile app.

- Device specific characteristics. These are characteristics that are related to the device on which the app is installed.
- Network specific checks
- App checks. These are things to check that have to do with functionality that is frequently used in an app.
- App User interface checks.
- Store specific checks.

ii) Performance Testing

Performance testing is a non-functional testing technique performed to determine the system parameters in terms of responsiveness and stability under various workloads. Performance testing measures the quality attributes of the system, such as scalability, reliability and resource usage

Performance Testing Techniques:

- Load Testing
- Spike Testing
- Stress Testing
- Endurance Testing

a) Load Testing

Load Testing is a type of Non functional testing. It is a type of software testing which is conducted to understand the behavior of the application under a specific expected load. Load testing is performed to determine a system's behavior[9] under both normal and at peak conditions. It helps to identify the maximum operating capacity of an application. The primary goal of a load testing is to define the maximum amount of work a system can handle without significant performance degradation.

Examples of Load Testing:

- Downloading a series of large files from the internet.
- Running multiple applications on a computer simultaneously
- Assigning many jobs to a printer in a queue.
- Writing and reading data to and from a hard disk continuously
- Subjecting a server to a large amount of traffic.

b) Spike Testing

Spike Testing is a mechanism of testing which means when in a web page frequent number of visitor access the page unexpectedly increases to maximum then obviously performance of the page breaks down. Spike testing is usually done by unexpectedly increasing the number of loads generated by users by a very enormous amount and observing the dramatic behavior of the system. The goal of spike testing is to regulate whether performance will deteriorate, the system will always fail, or it will be able to hold dramatic changes in load.   **Example:**
When we check the results on JNTUH site, site is suddenly loaded and unloaded and then the IT squad of JNTUH checks how the site reacts with unexpected increase and decrease of users.

c) Stress Testing

Stress testing is the process of determining the ability of a computer, network, program or device to maintain a certain level of effectiveness under unfavorable conditions.

Example: Handling 25 user login

If there is an application which can handle 25 simultaneous user login at a time.

- In stress testing we will test with more users than 25 and the test will continue to any number.

d) Endurance Testing

Endurance Testing is a type of performance testing which is usually used to determine how much a system can sustain the continuous expected load. During the period of Endurance testing memory utilization is always monitored to detect any potentials leaks.

Example: Banking Application

In closing days of bank we have continuous load on that days so we always test the banking application by keeping in mind the endurance testing.

iii) Compatibility Testing

Compatibility testing is a non functional testing conducted on the application to evaluate the application's compatibility within different environments.

Types of compatibility testing:

- Browser compatibility testing
- Hardware testing
- Networks
- Mobile devices
- Operating System
- Versions

Common Compatibility testing defects

- Changes in UI (look and feel)
- Change in font size
- Alignment related issues
- Change in style and color
- Scroll bar related issues
- Content or Label overlapping

    Example: ebay.com

For example to test the compatibility of site ebay.com. Download different versions of Firefox and install them one by one and test the eBay site.
EBay site should behave equally same in each version.

iv) Security Testing

Security Testing[7,8] is a type of software testing that intends to uncover vulnerabilities of the system and determine that its data and resources are protected from possible intruders. The six basic security concepts that need to be covered by security testing are:

- Confidentiality
- Integrity
- Authentication
- Authorization
- Non repudiation

Example: Web Application

The Security tester should have good knowledge of the HTTP protocol. It is important to have an understanding of how the client and the server communicate using HTTP. The tester should at least know the basics of SQL injection and XSS.

- Password Cracking

The Security testing on a web application can be kicked off by "password cracking". In order to login to the private areas of the application, one can either guess a username /password cracker tool for the same.

v) Data Volume Testing

Volume testing is a Non-functional testing that is performed as part of performance testing where the software is subjected to a huge volume of data. It is also referred as flood testing. If we expect certain database growth, we may want to artificially grow the database to that size and test the performance of the application when using it. System performance can degrade when large amounts of data must be searched or indexed. This kind of testing can determine the amount of data the application can handle before it starts to display errors or even stop responding. During this test, testing team operates software by storing sample data to estimate capacity of the software database.

Volume Testing Characteristics:

- During development phase, only small amount of data is tested.
- The performance of the software deteriorates over time as there is enormous amount of data overtime.
- Test cases are derived from design documents.
- Test data is usually generated using a test data generator.

- Test data need not be logically correct but the data is to assess the system performance.
- Upon completion of testing, results are logged and tracked to bring it to closure.

II.   CONCLUSION

Software testing is one of the important phases of software life cycle that aims to make the program error free and ensures product quality. The cost of testing is generally higher than the cost of remaining activities in the software development life cycle. Test planning is carried out before performing a test. There are two types of test cases design strategies black box and white box testing.

REFERENCES

[1] Pressman, R.S. 1997. Software Engineering: A practitioner Approach.4th Edition. Tata McGraw Hill.

[2] Sommerville, I. 1998. Software Engineering. 5th edition. Addison-Wesley.

[3] Myers, G.J. The Art of Software Testing.New York: John Wiley and Sons.

[4] McGraw.Chess,B.Seven pernicious kingdom.2005. A Taxonomy of Software Security Errors.WISST Workshop on Software Security Assurance Tools, Techniques and metrices.

[5] Sarma, M. D,Kundu.Mall, R. 2007. Automatic Test Case Generation from UML Sequence Diagram. International Conference on Advance Computing and Communication. Doi:10.1109/ADCOM:2007.68.

[6] Rina, DCSK, KU,Haryana ,INDIA,Tyagi,Sanjay.DCSA,KU,Haryana.2013.AComp arative Study Of Performance Testing Tools.IJARCSSE. 3,2(May 2013)

[7] Karen Scarfone.2012. Intro to Information Security Testing & Assessment. ScarfonecyberSecurity Csr.nist.gov.(7June 2012).

[8] B,Beizer .1990.Software Testing Techniques. Technology Maturation and Research Strategies Carneige Mellon University Pittsburg,USA.

[9] B.Beizer .1995. Software Testing Techniques.2006.Van Nostrand Reinhold, New York.1990.ISBN.0-442-20672-0.(31.Oct.2006).

[10] Khan,Mohd.Khan,Farmeena.2012. A Comparative Study of White Box, Black Box and Grey Box Testing Techniques.2012. International Journal of Advanced Computer Science and Applications(IJACSA). Vol. 3.No.6.(2012)

[11] Swain,S.k.Mohapatra,D.P.Mall, R.2010.Test Case Generation Based on Use Case and Sequence

Diagram.International journal of Software Engineering (IJSE) .3, 2.(July 2010),(289-321).

[12] hakre, Sheetal. Chavan, savita. Chavan, P.M. 2012.Software Testing Strategies and Techniques. International Journal of Emerging Technology and Advanced Engineering .Website: www.ijetae.com .2,4.(April 2012), (2250-2459).

Dr. B. V.Ramana Murthy has done his PhD from Osmania University, presently he working as Professor in Computer Science and Engineering, has 18 years of experience in Teaching and R&D. His primary area of interest is Software Engineering & Web Engineering.

Dr. V Padmakar has done his PhD in CSE, presently working as Associate Professor in the Department of Computer Science and Engineering has 18 years of experience in Teaching and Industry. His primary area of interests is Software Engineering, Network Security, and Data mining