International Journal of Trend in Research and Development, Volume 7(4), ISSN: 2394-9333
www.ijtrd.com

Levels of Testing

[1]Chetashri Bhusari, [2]Shonal Vaz and [3]Supriya Angne,
[1,2]Lecturer, Department of Information technology, Vidyalankar Polytechnic, Mumbai, India
[3]Department of computer Engineering technology, Vidyalankar Polytechnic, Mumbai, India

***Abstract*—**Testing is nothing but of find out an error. Types of testing and characteristics are represented in this paper. paper represents types and levels of testing.

Keywords*—Software Testing; Unit Testing*

I. INTRODUCTION

Software Testing is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is Defect free. It involves the execution of a software component or system component to evaluate one or more properties of interest. Software testing also helps to identify errors, gaps, or missing requirements in contrary to the actual requirements. It can be either done manually or using automated tools.

II. BENEFIT IS OF TESTING

Here are the benefits of using software testing:

Cost-Effective: It is one of the important advantages of software testing. Testing any IT project on time helps you to save your money for the long term. In case if the bugs caught in the earlier stage of software testing, it costs less to fix.

Security: It is the most vulnerable and sensitive benefit of software testing. People are looking for trusted products. It helps in removing risks and problems earlier.

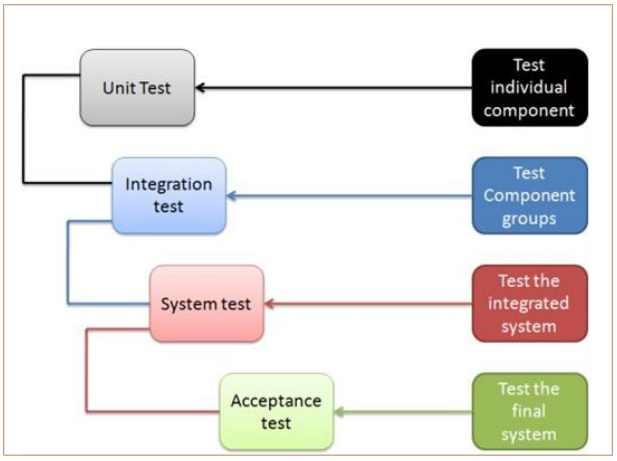
Product quality: It is an essential requirement of any software product. Testing ensures a quality product is delivered to customers.

Customer Satisfaction: The main aim of any product is to give satisfaction to their customers. UI/UX Testing ensures the best user experience.

III. TESTING STRATEGIES IN SOFTWARE ENGINEERING

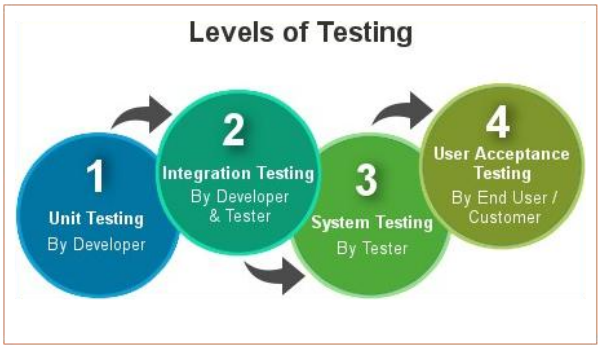
Here are important strategies in software engineering:

Unit Testing: This software testing approach is followed by the programmer to test the unit of the program. It helps developers to know whether the individual unit of the code is working properly or not.



Integration testing: It focuses on the construction and design of the software. You need to see that the integrated units are working without errors or not.

System testing: In this method, your software is compiled as a whole and then tested as a whole. This testing strategy checks the functionality, security, portability, amongst others.



IV. UNIT TESTING

Unit Testing is a level of software testing where individual units/ components of a software are tested. Purpose is to validate that each unit of the software performs as designed. Unit Testing is the first level of testing and is performed prior to Integration Testing. A unit is the smallest testable part of software. It is executed by the Developer.

Unit Testing is performed by using the White Box Testing method. Example: - A function, method, Loop or statement in program is working fine.

- The module interface is tested to ensure that information properly flows into and out of the program unit under test.
- Local data structures are examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm's execution.
- All independent paths through the control structure are exercised to ensure that all statements in a module have been executed at least once.
- Boundary conditions are tested to ensure that the module operates properly at boundaries established to limit or restrict processing.
- Error-handling paths Ensure algorithm responds correctly to specific error conditions.

Advantages of unit testing:

- Unit testing increases confidence in changing/ maintaining code.
- If good unit tests are written then we will be able to promptly
- catch any defects introduced due to the change.
- The cost of fixing a defect detected during unit testing is lesser in comparison to that of defects detected at higher levels.
- Debugging is easy

Drivers :

- It can simulate the behavior of upper-level module

that is not integrated yet.

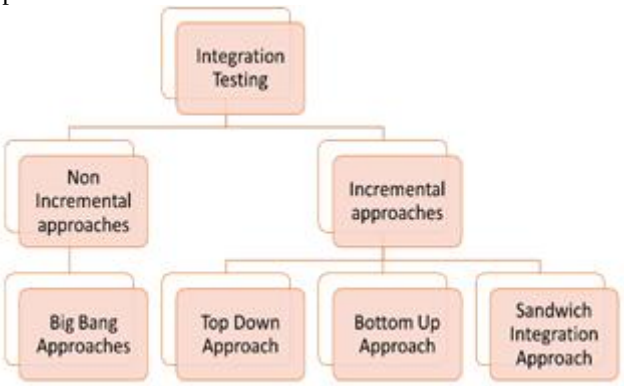
- Drivers modules act as the temporary replacement of module and act as the actual products.
- A simple main program that accepts test case data, passes such data to the component being tested, and prints the returned results

Stubs:

- Serve to replace modules that are subordinate to (called by) the component to be tested
- It can simulate the behavior of lower-level module that are not integrated.
- They are act as a temporary replacement of module and provide same output as actual product.

V. INTEGRATION TESTING

Integration Testing is a level of software testing where individual units are combined and tested as a group. After Unit Testing integration testing is done. Integration Testing Approaches.



Non incremental approach → Big Bang Approach

- Commonly called the "Big Bang" approach.
- All components are combined in advance.
- The entire program is tested as a whole.

When it is performed: After all unit are ready big bang integration is performed.

Advantages of Big Bang Approach: Convenient for small systems

Disadvantages of Big Bang Approach:

- Since all modules are tested at once, high risk critical modules are not isolated and tested on priority.
- Correction is difficult because isolation of causes is complicated.
- Once a set of errors are corrected, more errors occur, and testing appears to enter an endless loop
- Integration testing can commence only after "all" the modules are designed, testing team will have less time for execution in the testing phase.

Three kinds of Incremental approach:

- Top-down integration
- Bottom-up integration
- Sandwich integration

The program is constructed and tested in small increments.

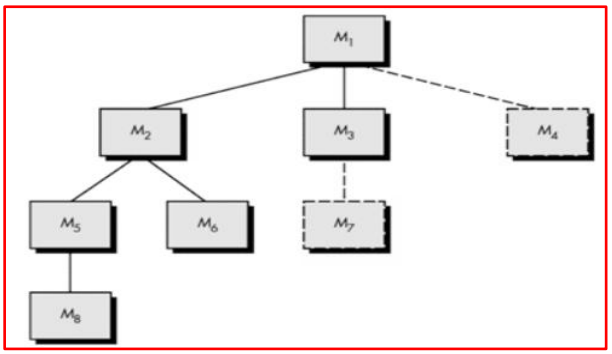
- Errors are easier to isolate and correct.
- Interfaces are more likely to be tested completely.
- A systematic test approach is applied

Incremental approach→ Top-down integration

- Modules are integrated by moving downward through the control hierarchy, beginning with the main module.
- It takes help of dummy program called stub for testing.
- Subordinate modules are incorporated in either a depth-first or breadth-first fashion.

Integration can be done in two ways:

- Depth First Method: All modules on a major control path are integrated.
- Breadth First method: All modules directly subordinate at each level are integrated.



Incremental approach: Top-down integration procedure

1. Main control module used as a test driver and stubs are substitutes for components directly subordinate to it.
2. Subordinate stubs are replaced one at a time with real components. (following the depth-first or breadth-first approach).
3. Tests are conducted as each component is integrated.
4. On completion of each set of tests and other stub is replaced with a real component.
5. Regression testing may be used to ensure that new errors not introduced.

Advantages of Top-down integration

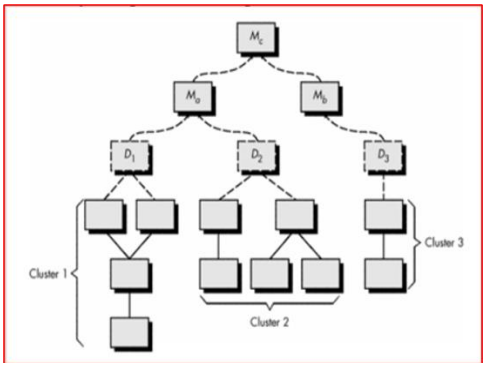
- Fault Localization is easier.
- Possibility to obtain an early prototype.
- Critical Modules are tested on priority; major design flaws could be found and fixed first.

Disadvantages of Top-down integration

- Stubs need to be created to substitute for modules that have not been built or tested yet; this code is later discarded.

Incremental approach→ Bottom Up integration

- Modules are integrated by moving upward through the control hierarchy, beginning with the lower level module.
- It takes help of dummy program called driver for testing.



Incremental approach→ Bottom up integration

- Low level components are combined in clusters that perform a specific software function.
- A driver (control program) is written to coordinate test case input and output

Incremental approach→ Bottom up integration procedure

- Drivers are written.
- The cluster is tested.
- Drivers are removed.
- Clusters are combined moving upward in the program structure.

Advantages of Bottom Up integration

- This approach verifies low-level data processing early in the testing process.
- Need for stubs is eliminated

Disadvantages of Bottom Up integration

- Driver modules need to be built to test the lower-level modules.
- This code written for drivers is later discarded or expanded into a full-featured version

Incremental approach→Sandwich integration

- Consists of a combination of both top-down and bottom-up integration.
- Occurs both at the highest level modules and also at the lowest level modules.
- Proceeds using functional groups of modules, with each group completed before the next.
- High and low-level modules are grouped based on the control and data processing they provide for a specific program feature.
- Integration within the group progresses in alternating steps between the high and low level modules of the group.
- When integration for a certain functional group is complete, integration and testing moves onto the next group.
- Incremental approach→Sandwich integration

Advantages:

- Repeats the advantages of both types of integration.

Disadvantages

- Requires a disciplined approach so that integration doesn't tend towards the "big bang" scenario.
- Each new addition or change to baselined software may cause problems with functions that previously worked flawlessly.
- Regression testing re-executes a small subset of tests that have already been conducted
- Ensures that changes have not propagated unintended side effects
- Helps to ensure that changes do not introduce unintended behavior or additional errors
- May be done manually or through the use of automated capture/playback tools
- Regression test suite contains three different classes of test cases
- A representative sample of tests that will exercise all software functions.
- Additional tests that focus on software functions that are likely to be affected by the change.
- Tests that focus on the actual software components that have been changed.

Client Server Testing

- Client server testing gives an opportunity for number of users to work with software at time.
- In simple terms request are coming from number of clients for doing some actions and server is serving these request.

Testing approaches of client server system -→Component Testing

- For testing Client and server individually approach and test plan need to be defined.
- One may have to devise simulators to replace corresponding components while testing the component targeted by the test.
- When server is tested, we may need a client simulator, while testing of client may need server simulator.

Testing approaches of client server system -→ Integration Testing

- After successful testing of servers, clients and network, they are brought together to form the system and system test cases are executed.
- Communication between client and server is tested in integration testing.

Testing approaches of client server system -→Performance Testing

- System performance is tested when number of clients are communicating with server at a time.
- we can test the system under maximum load as well as normal load expected.

Testing approaches of client server system -→ Concurrency Testing

- It may be possible that multiple users may be accessing same record at a time.
- Concurrency testing is required to understand the behavior of a system under such circumstances.

Testing approaches of client server system -→Disaster Recovery Testing

- When the client and server are communicating with each other, there exists a possibility of breaking of the communication due to various reasons or failure of either client or server or link connecting them.
- It may involve testing the scenario of such failure at different points in the system and action taken by the system in each case.

Testing approaches of client server system -→Testing for extended periods

- In client server application it may be expected that server is running 24*7 for extended period.
- one need to conduct testing over an extended period to understand if service level of network and server deteriorates over time due to some reasons.

Testing approaches of client server system -→Compatibility Testing

- Servers may be in different hardware, software or operating environment than the recommended one.
- Client may differ significantly from the expected environmental variables.
- Testing must ensure that performance must be maintained on the range of hardware and software configurations.

Web Application Testing

- Web application is further improvement in client server applications.
- The clients can communicate with server through virtual connectivity.

Testing approaches of web application testing-→Component Testing

- For testing web application approach and test plan need to be defined.
- One may have to devise simulators to replace corresponding components while testing the component targeted by the test.
- When server is tested, we may need a client simulator, while testing of client may need server simulator.

Testing approaches of web application testing-→Integration Testing

- After successful testing of servers, clients and network, they are brought together to form the system and system test cases are executed.
- Communication between client and server is tested in integration testing.

Testing approaches of web application testing-→Performance Testing

- System performance is tested when number of clients are communicating with server at a time.
- We can test the system under maximum load as well as normal load expected.

Testing approaches of web application testing-→Concurrency Testing

- It may be possible that multiple users may be accessing same record at a time.
- Concurrency testing is required to understand the behavior of a system under such circumstances.

Testing approaches of web application testing-→ Disaster Recovery Testing

- When machine and web server are communicating with each other, there exists a possibility of breaking of the communication due to various reasons or failure of either client or server or link connecting them.
- It may involve testing the scenario of such failure at different points in the system and action taken by the system in each case.

Testing approaches of web application testing-→ Testing for extended period

- In web application it may be expected that server is running 24*7 for extended period.
- one need to conduct testing over an extended period to understand if service level of network and server deteriorates over time due to some reasons.

Testing approaches of web application testing-→Compatibility Testing

- Servers may be in different hardware, software or operating environment than the recommended one.
- Client browser may differ significantly from the expected environmental variables.
- Testing must ensure that performance must be maintained on the range of hardware and software configurations.

Testing approaches of web application testing-→Security Testing

- As the communication is through virtual network, security becomes an important issue.
- Application may use communication protocols, coding and decoding mechanism and schemes to maintain security of system.
- System must be tested for possible weak areas called vulnerabilities and possible intruders trying to attack the system.

Performance Testing

- Performance testing is intended to find whether the system meets its performance requirements under normal load or abnormal level of activities.
- Normal load must be defined by the requirement statement defined by the customer and system design implements them.
- Performance criteria must be expressed in numerical terms.
- This is one area where verification does not work to that much extents and one needs to test it by actually performing the operation on the system.

Load Testing

- Load testing is a type of non-functional testing.
- A load test is 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 under both normal and at peak conditions.
- It helps to identify the maximum operating capacity of an application as well as any bottlenecks and determine which element is causing degradation.
- Example : If the software operates on peripherals such as printer, or communication ports, connect as many as you can.
- Example: If you are testing an internet server that can handle thousands of simultaneous connections, do it.
- Scenario like, most software it is important for it to run over long periods.
- Some software's should be able to run forever without being restarted.

Stress Testing

- It is a type of non-functional testing.
- Stress testing is testing the software under less than ideal conditions.
- So subject your software to low memory, low disk space, slow CPU, and slow modems and so on.
- Look at your software and determine what external resources and dependencies it has.
- Stress testing is simply limiting them to bare minimum. With stress testing you starve the software.
- The goals of such tests may be to ensure the software does not crash in conditions of insufficient computational resources (such as memory or disk space).

Security Testing

- Verifies that protection mechanisms built into a system will, in fact, protect it from improper access.
- As the communication is through virtual network, security becomes an important issue.
- Application may use communication protocols, coding and decoding mechanism and schemes to maintain security of system.

- System must be tested for possible weak areas called vulnerabilities and possible intruders trying to attack the system.

VI. ACCEPTANCE TESTING

- Final stage of testing.
- Generally done by end user /customer/third party/system testing people.
- Required document: Test plan/Project plan which defines acceptance criteria.

Acceptance testing validates following:

- Whether user needs defined in SRS are satisfied by System or not .
- Whether system meets performance requirement as specified in SRS.
- It also determines whether the application satisfies its acceptance criteria

Acceptance testing has two types:

Alpha testing

- Testing is done by end user at developer site.
- It is done in natural setting of software.
- Developer is present.
- Over the shoulder approach.
- Usage related error recorded.
- Controlled environment.

Beta Testing

Aim: Developer can release error free software to customer immediately.

- Testing is done by end user at end user site.
- It is live kind of application testing.
- Developer is not present
- Not done in controlled environment.
- End user get an opportunity to record error.
- Error reporting on day today basis.
- Error those were reported on daily basis get resolved immediately/some times fixed in next release.

IV. DIFFERENCE BETWEEN ALPHA TESTING AND BETA TESTING

Alpha Testing	Beta Testing
Testing is done by end user at developer site.	Testing is done by end user at end user site.
It is done in controlled environment.	Developer is not present so not done in controlled environment.
Alpha testing is not effective as it is done in controlled environment.	Beta test is more effective as end user do not have any restriction of testing software.
End user may have to test software under influence of developer.	Beta testing is live application testing, not done in influence of developer.

CONCLUSION

Testing is a critically important verification method that takes up a very large portion of a project's resources, including schedule, budget, staffing, and facilities. Unlike the many constructive activities of systems engineering, testing is relatively unique because it is inherently destructive.

References

[1] https://www.oreilly.com/library/view/common-system)
[2] https://www.infosys.com/it-services/validation-solutions/white-papers/documents/test-result-reporting.pdf.
[3] https://www.tutorialspoint.com/software_testing_dictionary/acceptance_testing.htm