

## ST Assignment No. 4

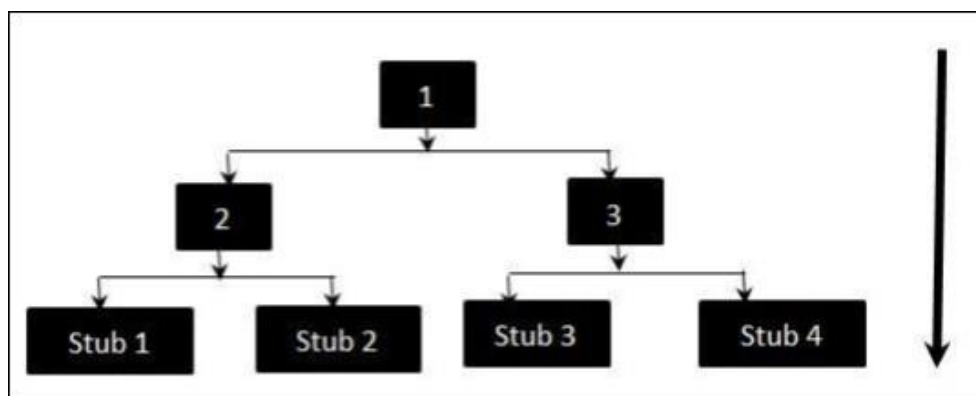
### Q1. Explain the concept of Stubs and Drivers in Unit Testing?

**Ans.**

- 1. Unit is the smallest testable part of the software system.*
- 2. Unit testing is done to verify that the lowest independent entities in any software are working fine.*
- 3. The smallest testable part is isolated from the remainder code and tested to determine whether it works correctly.*
- 4. When developer is coding the software it may happen that the dependent modules are not completed for testing, in such cases developers use stubs and drivers to simulate the called (stub) and caller (driver) units.*
- 5. Unit testing requires stubs and drivers, stubs simulate the called unit and driver simulates the calling unit.*

#### **1. STUBS:**

- i. Assume you have 3 modules, Module A, Module B and module C.*
- ii. Module A is ready and we need to test it, but module A calls functions from Module B and C which are not ready, so developer will write a dummy module which simulates B and C and returns values to module A.*
- iii. This dummy module code is known as stub.*



- iv. The above diagrams clearly states that Modules 1, 2 and 3 are available for integration, whereas, below modules are still under development that cannot be integrated at this point of time.*
- v. Hence, Stubs are used to test the modules.*

## **2. DRIVERS:**

i. Now suppose you have modules B and C ready but module A which calls functions from module B and C is not ready so developer will write a dummy piece of code for module A which will return values to module B and C.

ii. This dummy piece of code is known as driver.

## **Q2. Explain Top-Down and Bottom-Up integration testing technique with advantages and disadvantages.**

**Ans.**

### **i. Top down Testing:**

In this approach testing is conducted from main module to sub module.

ii. If the sub module is not developed a temporary program called STUB is used for simulate the sub module.

### **Advantages:**

- Advantageous if major flaws occur toward the top of the program.
- Once the I/O functions are added, representation of test cases is easier.
- Early skeletal Program allows demonstrations and boosts morale.

### **Disadvantages:**

- Stub modules must be produced
- Stub Modules are often more complicated than they first appear to be.
- Before the I/O functions are added, representation of test cases in stubs can be difficult.
- Test conditions may be impossible, or very difficult, to create.
- Observation of test output is more difficult. □ Allows one to think that design and testing can be overlapped.
- Induces one to defer completion of the testing of certain modules.

### ***i. Bottom up testing:***

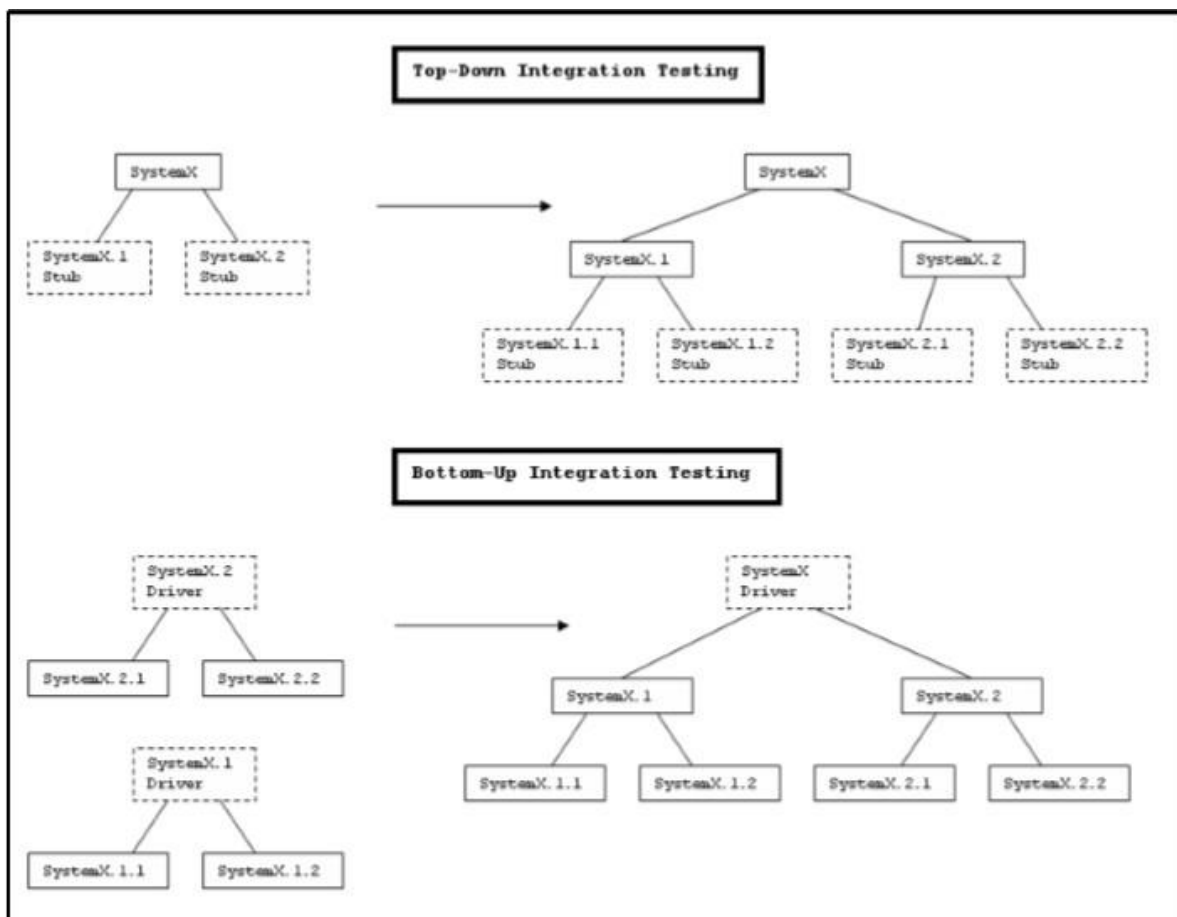
*In this approach testing is conducted from sub module to main module, if the main module is not developed a temporary program called DRIVERS is used to simulate the main module.*

#### **Advantages:**

- *Advantageous if major flaws occur toward the bottom of the program.*
- *Test conditions are easier to create. □ Observation of test results is easier.*

#### **Disadvantages:**

- *Driver Modules must be produced.*
- *The program as an entity does not exist until the last module is added.*



### **Q3. Describe incremental integration testing with features.**

**Ans.**

- *After unit testing is completed, developer performs integration testing.*
- *It is the process of verifying the interfaces and interaction between modules.*
- *While integrating, there are lots of techniques used by developers and one of them is the incremental approach.*
- *In Incremental integration testing, the developers integrate the modules one by one using stubs or drivers to uncover the defects.*
- *This approach is known as incremental integration testing.*
- *To the contrary, big bang is one other integration testing technique, where all the modules are integrated in one shot.*

#### **Features**

- Each Module provides a definitive role to play in the project/product structure*
- Each Module has clearly defined dependencies some of which can be known only at the runtime.*
- The incremental integration testing's greater advantage is that the defects are found early in a smaller assembly when it is relatively easy to detect the root cause of the same.*
- A disadvantage is that it can be time-consuming since stubs and drivers have to be developed for performing these tests.*

### **Q4. Explain System Testing in details and explain any 4 system testing approaches.**

**Ans.**

- System Testing (ST) is a black box testing technique performed to evaluate the complete system the system's compliance against specified requirements.*
- In System testing, the functionalities of the system are tested from an end-to-end perspective.*
- System Testing is usually carried out by a team that is independent of the development team in order to measure the quality of the system unbiased.*
- It includes both functional and Non-Functional testing.*

#### **1. Recovery Testing**

- Recovery testing is a type of non-functional testing technique performed in order to determine how quickly the system can recover after it has gone through system crash or hardware failure.*

*ii. Recovery testing is the forced failure of the software to verify if the recovery is successful.*

**Steps:**

- *Determining the feasibility of the recovery process.*
- *Verification of the backup facilities.*
- *Ensuring proper steps are documented to verify the compatibility of backup facilities.*
- *Providing Training within the team.*
- *Demonstrating the ability of the organization to recover from all critical failures.*
- *Maintaining and updating the recovery plan at regular intervals.*

## **2. Security Testing**

*i. Security testing is a testing technique to determine if an information system protects data and maintains functionality as intended.*

*ii. It also aims at verifying 6 basic principles as listed below:*

- *Confidentiality*
- *Integrity*
- *Authentication*
- *Authorization*
- *Availability*
- *Non-repudiation*

**Techniques:**

- *Injection*
- *Broken Authentication and Session Management*
- *Cross-Site Scripting (XSS)*
- *Insecure Direct Object References*
- *Security Misconfiguration*

- *Sensitive Data Exposure*
- *Missing Function Level Access Control*
- *Cross-Site Request Forgery (CSRF)*
- *Using Components with Known Vulnerabilities*
- *Invalidated Redirects and Forwards*

### **3. Load Testing.**

*i. Load testing is performance testing technique using which the response of the system is measured under various load conditions.*

*ii. The load testing is performed for normal and peak load conditions.*

#### **Load Testing Approach:**

- *Evaluate performance acceptance criteria*
- *Identify critical scenarios*
- *Design workload Model*
- *Identify the target load levels*
- *Design the tests*
- *Execute Tests*
- *Analyse the Results*

#### **Objectives of Load Testing:**

- *Response time*
- *Throughput*
- *Resource utilization*
- *Maximum user load*
- *Business-related metrics*

### **4. Compatibility Testing.**

*i. Compatibility testing is a non-functional testing conducted on the*

*application to evaluate the application's compatibility within different environments.*

- ii. It can be of two types - forward compatibility testing and backward compatibility testing.*
- Operating system Compatibility Testing - Linux , Mac OS, Windows*
- Database Compatibility Testing - Oracle SQL Server*
- Browser Compatibility Testing - IE , Chrome, Firefox*
- Other System Software - Web server, networking/ messaging tool, etc*

### **Q5. Explain criteria for acceptance testing.**

**Ans.**

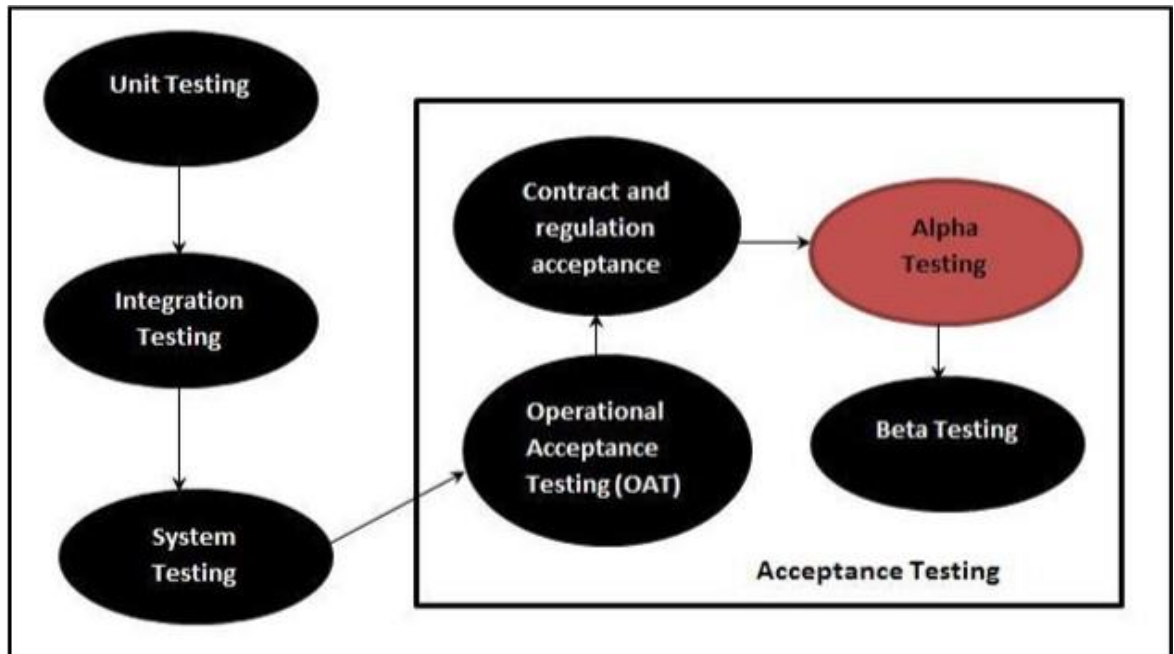
- i. Comparison testing comprises of comparing the contents of files, databases, against actual results.*
- ii. They are capable of highlighting the differences between expected and actual results.*
- iii. Comparison test tools often have functions that allow specified sections of the files be ignored or masked out.*
- iv. This enables the tester to mask out the date or time stamp on a screen or field as it is always different from the expected ones when a comparison is performed.*

### **Q6. Explain Alpha Testing and Beta Testing.**

**Ans.** *Alpha testing takes place at the developer's site by the internal teams, before release to external customers. This testing is performed without the involvement of the development teams.*

- i. Alpha Testing - In SDLC*

*The following diagram explains the fitment of Alpha testing in the software development life cycle.*



## ii. *How do we run it?*

- *In the first phase of alpha testing, the software is tested by in-house developers during which the goal is to catch bugs quickly.*
- *In the second phase of alpha testing, the software is given to the software QA team for additional testing.*
- *Alpha testing is often performed for Commercial off-the-shelf software (COTS) as a form of internal acceptance testing, before the beta testing is performed.*

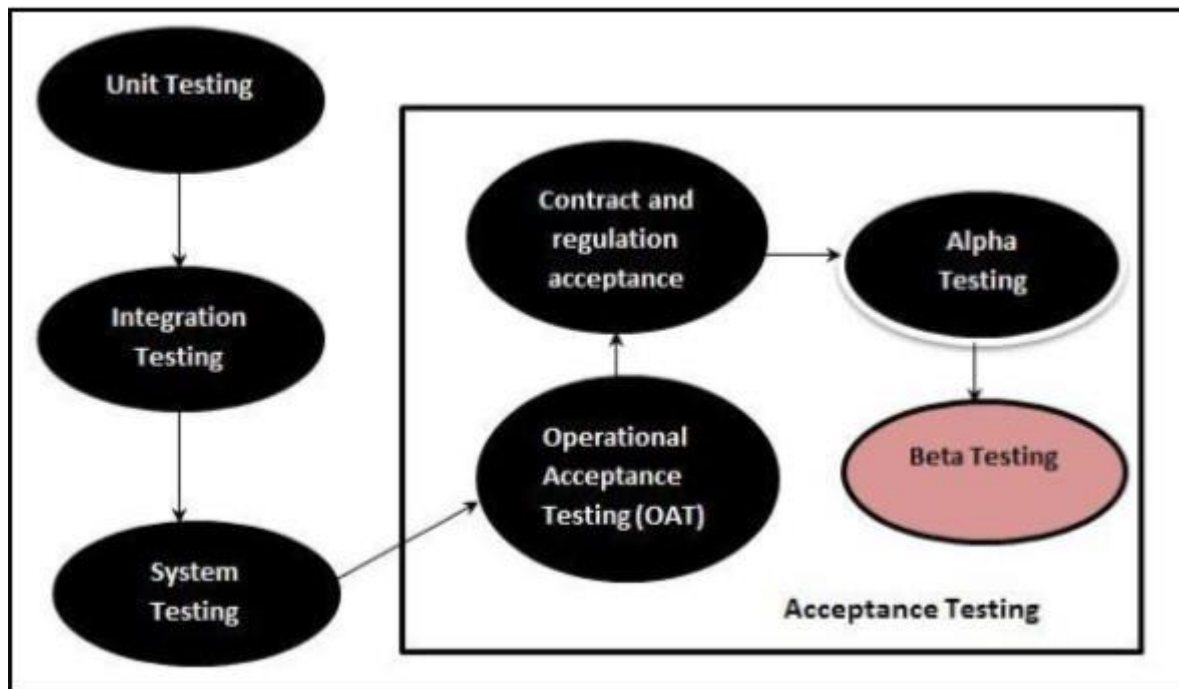
## **Beta Testing:**

- *Beta testing also known as user testing takes place at the end users site by the end users to validate the usability, functionality, compatibility, and reliability testing.*
- *Beta testing adds value to the software development life cycle as it allows the "real" customer an opportunity to provide inputs into the design, functionality, and usability of a product. These inputs are not only critical to the success of the product but also an investment into future products when the gathered data is managed effectively.*
- *Beta Testing - In SDLC*

*The following diagram explains the fitment of Beta testing in the software development*



life cycle:



### Q7. Explain Object oriented application Testing.

**Ans.**

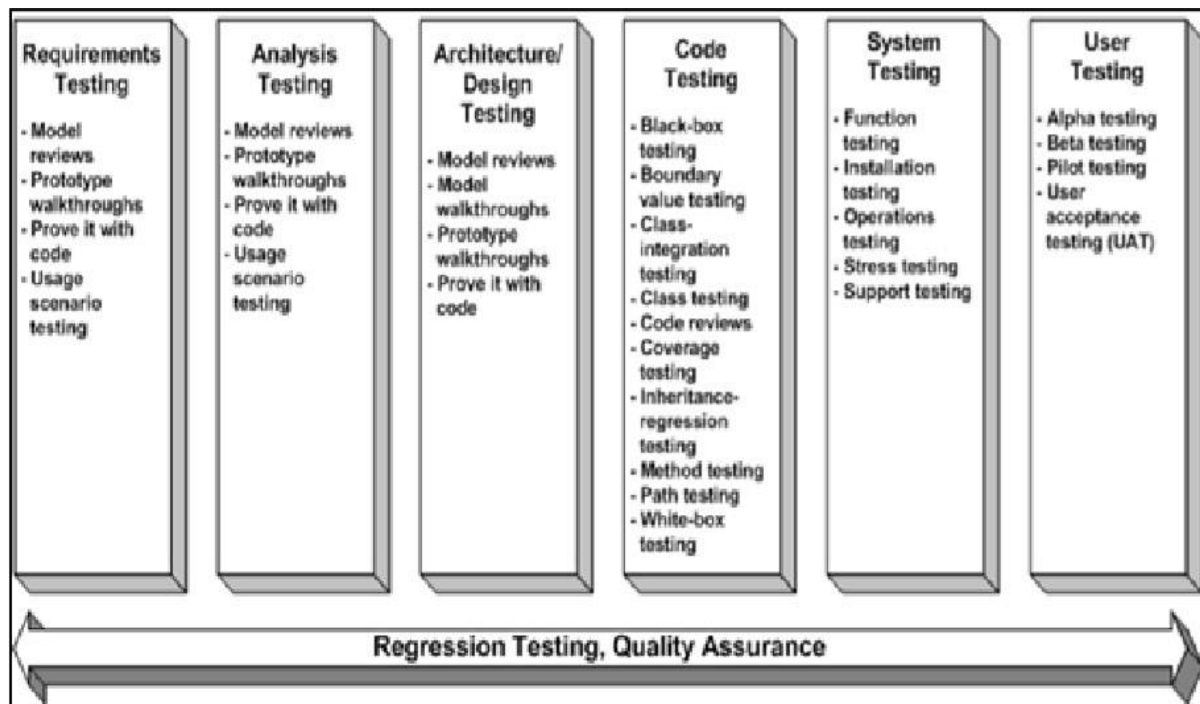
*The Full-Lifecycle Object-Oriented Testing (FLOOT) methodology is a collection of testing techniques to verify and validate object-oriented software.*

*The FLOOT lifecycle is depicted in Figure 9, indicating a wide variety of techniques (described in Table 9) are available to you throughout all aspects of software development.*

*The list of techniques is not meant to be complete: instead the goal is to make it explicit that you have a wide range of options available to you.*

*It is important to understand that although the FLOOT method is presented as a collection of serial phases it does not need to be so: the techniques of FLOOT can be applied with evolutionary/agile processes as well.*

*The reason why I present the FLOOT in a "traditional" manner is to make it explicit that you can in fact test throughout all aspects of software development, not just during coding.*



**Q8. Explain Client server testing in terms of your final year project.**

**Ans.**

- i. This type of testing usually done for 2 tier applications (usually developed for LAN) Here we will be having front-end and backend.*
- ii. The application launched on front-end will be having forms and reports which will be monitoring and manipulating data. E.g: applications developed in VB, VC++, Core Java, C, C++, D2K, PowerBuilder etc.,*
- iii. The backend for these applications would be MS Access, SQL Server, Oracle, Sybase, Mysql, Quadbase.*
- iv. The tests performed on these types of applications would be– User interface testing  
Manual support testing– Functionality testing– Compatibility testing & configuration testing – Intersystem testing.*

**Q9. Describe various web application testing techniques.**

**Ans.**

**1. Functionality Testing** - The below are some of the checks that are performed but not limited to the below list:

- Verify there is no dead page or invalid redirects.
- First check all the validations on each field.
- Wrong inputs to perform negative testing.
- Verify the workflow of the system.
- Verify the data integrity.

**2. Usability testing** - To verify how the application is easy to use with.

- Test the navigation and controls.
- Content checking.
- Check for user intuition.

**3. Interface testing** - Performed to verify the interface and the data flow from one system to other.

**4. Compatibility testing** - Compatibility testing is performed based on the context of the application.

- Browser compatibility
- Operating system compatibility
- Compatible to various devices like notebook, mobile, etc.

**5. Performance testing** - Performed to verify the server response time

*and throughput under various load conditions.*

- *Load testing - It is the simplest form of testing conducted to understand the behaviour of the system under a specific load. Load testing will result in measuring important business critical transactions and load on the database, application server, etc. are also monitored.*
- *Stress testing - It is performed to find the upper limit capacity of the system and also to determine how the system performs if the current load goes well above the expected maximum.*
- *Soak testing - Soak Testing also known as endurance testing, is performed to determine the system parameters under continuous expected load. During soak tests the parameters such as memory utilization is monitored to detect memory leaks or other performance issues. The main aim is to discover the system's performance under sustained use.*
- *Spike testing - Spike testing is performed by increasing the number of users suddenly by a very large amount and measuring the performance of the system. The main aim is to determine whether the system will be able to sustain the work load.*

**6. Security testing** - *Performed to verify if the application is secured on web as data theft and unauthorized access are more common issues and below are some of the techniques to verify the security level of the system.*

- *Injection*
- *Broken Authentication and Session Management*
- *Cross-Site Scripting (XSS)*
- *Insecure Direct Object References*
- *Security Misconfiguration*
- *Sensitive Data Exposure*
- *Missing Function Level Access Control*
- *Cross-Site Request Forgery (CSRF)*
- *Using Components with Known Vulnerabilities*
- *Invalidated Redirects and Forwards*

### **Q10. Explain Performance Testing with example.**

**Ans.**

**Performance testing** - Performed to verify the server response time and throughput under various load conditions.

- *Load testing* - It is the simplest form of testing conducted to understand the behavior of the system under a specific load. Load testing will result in measuring important business critical transactions and load on the database, application server, etc. are also monitored.
- *Stress testing* - It is performed to find the upper limit capacity of the system and also to determine how the system performs if the current load goes well above the expected maximum.
- *Soak testing* - Soak Testing also known as endurance testing, is performed to determine the system parameters under continuous expected load. During soak tests the parameters such as memory utilization is monitored to detect memory leaks or other performance issues. The main aim is to discover the system's performance under sustained use.
- *Spike testing* - Spike testing is performed by increasing the number of users suddenly by a very large amount and measuring the performance of the system. The main aim is to determine whether the system will be able to sustain the work load.