

## SOFTWARE TESTING

Q Explain STLC (Software testing lifecycle)? / strategic approach for SW testing.

⇒

- STLC is a sequence of activities during the testing process. Following are the phases in STLC model.

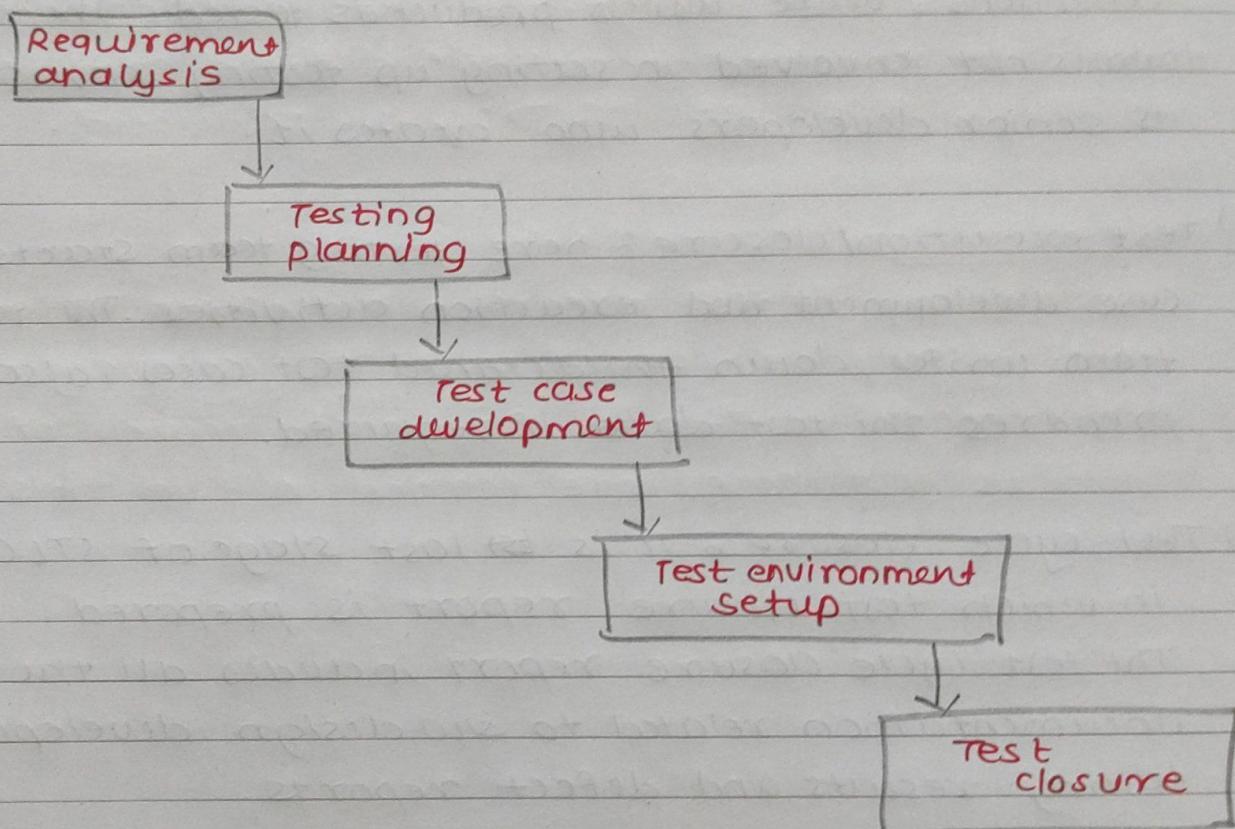


fig. STLC

### (i) Requirement Analysis :

- Identifying that the requirements are testable or not
- If requ. are not testable then team communicates to various stakeholders such as business analyst, clients, technical leads so that mitigation strategy can be planned

### (ii) Testing planning :

- first phase where senior QA manager determines the test plan. In this manager calculates estimated efforts and cost for testing work. Activities like resource planning, tool selection, determining roles, etc are carried out.

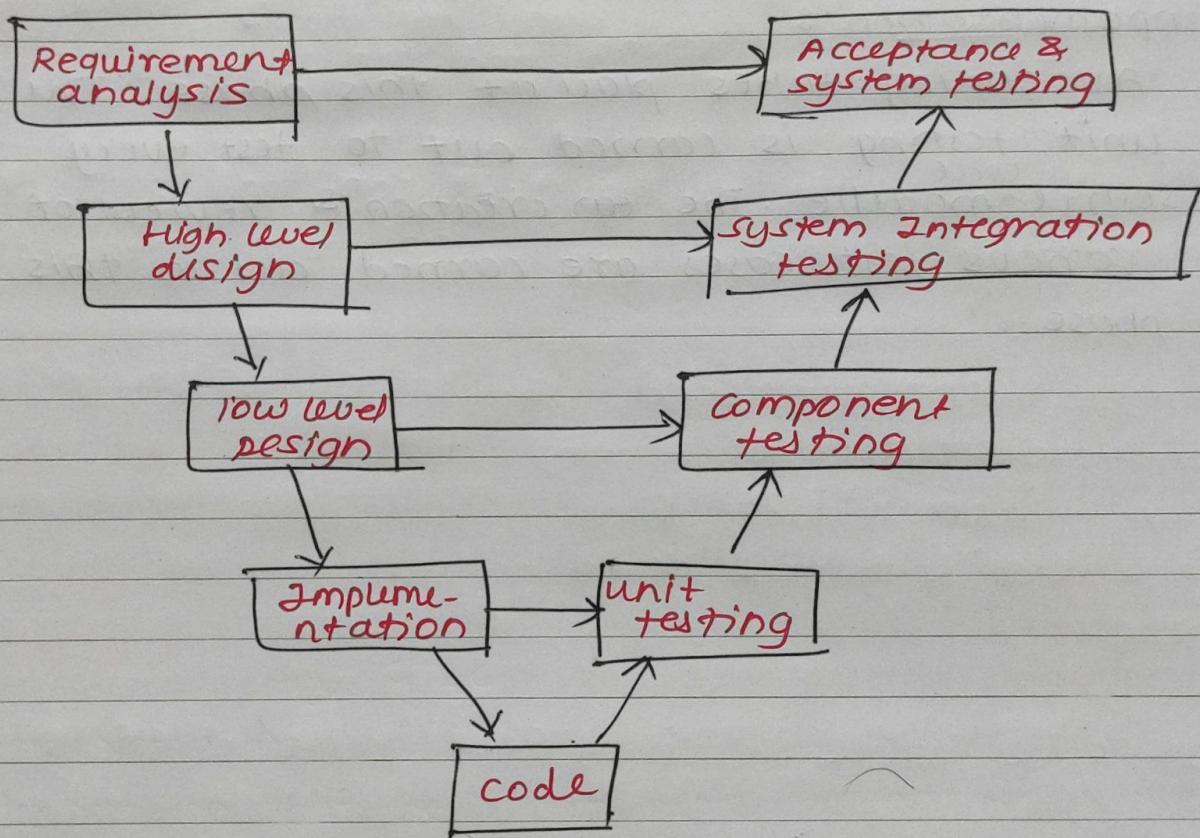
(iii) Test case development or design : Once testing planning is done test case development plan starts. In this phase , the test cases are created and the test data is identified

(iv) Test environment setup : It decides the SW & HW conditions under which product is tested . The testing team is not involved in setting up testing environment its senior developers who creates it.

(v) Test execution/closure : here testing team starts case development and execution activities . The testing team writes down the detailed test cases , also prepares the test data if required.

(vi) Test cycle closure : It is the last stage of STLC in which test closure report is prepared . The test cycle closure report includes all the documentation related to SW design , development , testing results and defect reports.

- ② Explain phases in verification and validation model
- - This model is V-shaped hence is the name. It is just like waterfall model in which each phase must be completed before the next phase starts.
  - This model contains various stages of SW development along with testing that can be conducted on each stage. Test plans serve as a link b/w development and testing stage.



- Requirement analysis: As per waterfall model the V model begins with requirement gathering and analysis. The SRS document is created during this phase. The acceptance and system testing is conducted in which focus is on meeting of functionalities specified in requirements gathering.

High level design: system architecture is designed.

It provide the overview of platform used, system.  
An integration test plan is ~~used~~ prepared and testing is conducted

Low level design: During this phase actual software components are designed. The relationship with other components are defined at this level. Here component testing is conducted.

Implementation:

All coding takes place at this phase. The unit testing is carried out to test every source code module. The ~~code~~ creation & review of various test cases are carried out in this phase.

## WHITE BOX TESTING

## BLACK BOX TESTING

- (i) Internal structure is being known to tester who is going to test SW.
  - (ii) This type of testing is carried out of SW developers.
  - (iii) Implementation knowledge is required.
  - (iv) Programming knowledge is required.
  - (v) Testing is applicable on lower level of testing like unit testing, Integration testing.
  - (vi) Also means structural or interior testing.
  - (vii) testing can be started based on Requirement specifications documents.
  - (viii) eg. functional testing, behavior testing, close box testing.
- (i) Test the software without knowing the internal structure of code.
  - (ii) This type of testing is carried out by testers.
  - (iii) Implementation knowledge is not required.
  - (iv) Programming knowledge is not required.
  - (v) Testing is applicable on higher level of testing like System and acceptance testing.
  - (vi) Also means functional or external testing.
  - (vii) testing can be started based on Detail design documents.
  - (viii) Eg. structural testing, logic testing, open box testing.



### VERIFICATION

1. The verification process include checking document, design, code and program.

2. It does not involve execution of code.

3. Verification uses method like reviews, walkthrough & inspection, desk checking.

4. It finds bugs early in the development cycle.

5. Target is application and software architecture, Specification complete design high level & database design etc

6. It comes before validation.

7. It involves Quality Assurance (QA) Team

8. It prevents errors

9. Done by Developer's

10. Static Nature

### VALIDATION

1. It is a dynamic technique of testing and validating the actual product.

2. It always involves execution of code.

3. It uses methods like Black Box Testing, White Box Testing and non-functional Testing.

4. It can find bugs that the verification process cannot catch.

5. Target is an actual product.

6. It comes after verification.

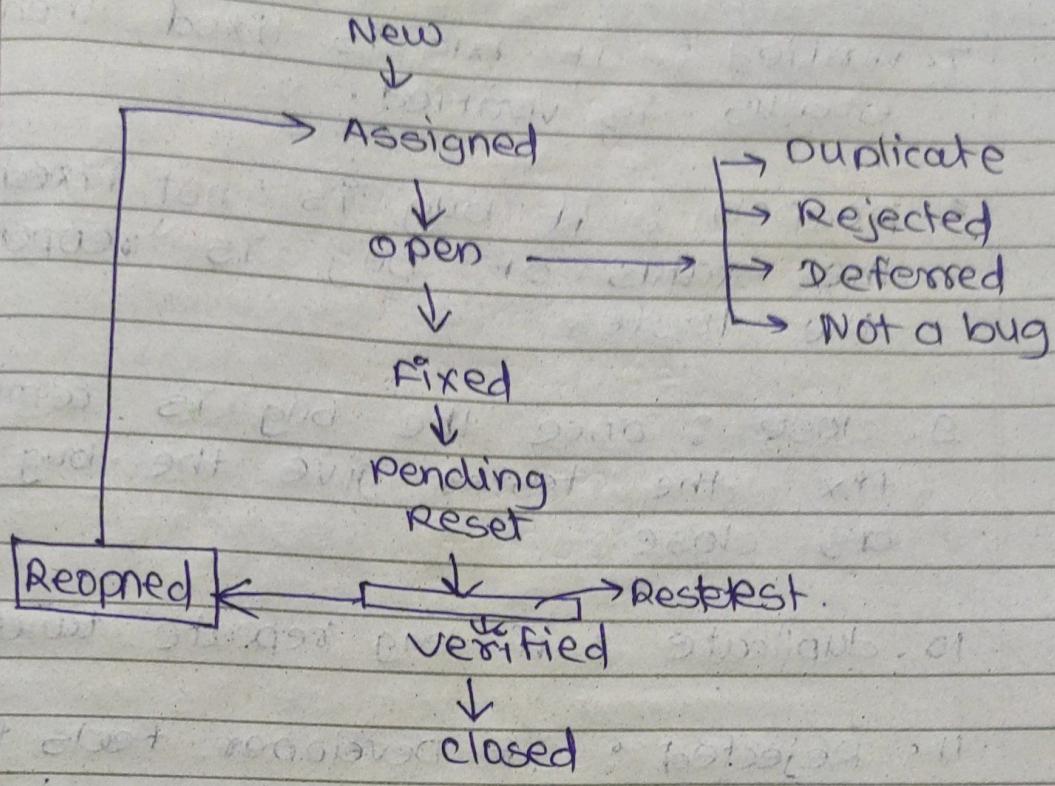
7. It involves Testing Team.

8. It detects errors.

9. Done by Testers.

10. Dynamic Nature.

## \* Defect life cycle.



1. **New** : when a defect is logged or posted
2. **Assigned** : After the tester has posted the bug the lead of the tester approve the bug and Assign to developer
3. **open** : developer start Analysing
4. **fixed** : when developer make some code changes and verified changes so the developer Make the bug state as 'fixed' passed to testing team
5. **Pending Reset** : After changing the code developer give the bug to tester and the state of bug pending Reset because it is pending for Re testing.

6. Retest : at this state tester do retest.

7. verified : if bug is fixed then it status is verified.

8. Reopen : if bug is not fixed then the status of bug is "reopen" by the tester

9. close : once the bug is completely fix the tester give the bug status as close.

10. duplicate : if bug repeat twice.

11. Rejected : if developer feels that bug is not genuine.

12. Deferred : bug is expected to fix in next release.  
- It depend on bug's priority.

13. Not a bug : if there is no change in application functionality then state is Not a bug.

### \* Verification :-

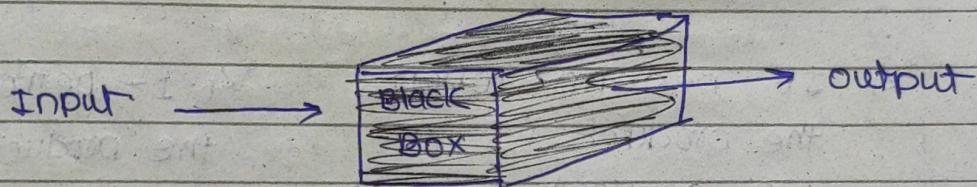
- verification is the process of checking that software achieve goal without any bugs
- static testing.
- It include documents, design, code
- Not involve execution of code.
- verification is done before the validation.
- It uses method like, inspections, do review, decision making etc.
- the Quality Assurance (QA) team would be engaged.
- It catch errors that validation cannot catch it is low level exercise.

### \* validation :-

- validation process include testing and validation of the actual product.
- Dynamic Testing
- It required execution the code to test functionality of product.
- use method like black box testing and non functional testing
- It come after verification.
- It catch errors that verification can not catch it is high level exercise.

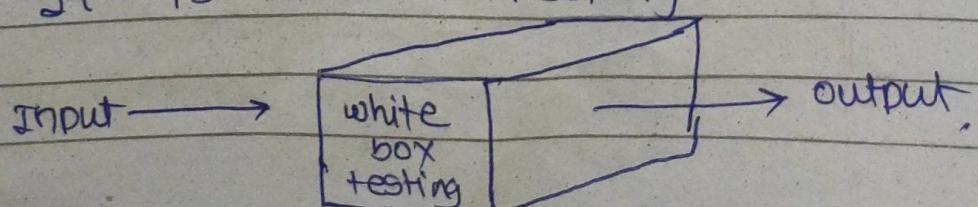
## \* Black - Box Testing .

- Technical documentation only no access to the code or information on the architecture of a tested product.
- It focus on software behaviour and functionality.
- Implementation knowledge is not required.
- Testing is applicable on higher levels of testing like system testing , Acceptance testing .
- programmer knowledge is not required
- It is external Testing



## \* white - Box testing :

- white box testing is the software testing method in which internal structure is being known to tester who is going to test the software.
- Implementation knowledge is required to perform white box testing.
- It focus on Internal logic of the software
- Testing is applicable on lower level of testing like unit testing or integration testing
- It is internal testing



## Alpha Testing

1. the employee of the companies perform Alpha testing
2. Alpha testing performed at the developer's site
3. It uses both white - box and blackbox testing
4. It perform when the product is 70-90 percent complete
5. perform in lab testing environment
6. done after system testing
7. Developers are engaged in process

## Beta testing

1. Beta testing is performed by the end user who are not the employees of the companies.
2. Beta testing is performed at the end-users place.
3. It uses black box testing technique
4. It perform when the product is 90-95 percent complete.
5. perform at the real time Environment.
6. done after Alpha testing
7. Developers not engaged in process.

## \* System Testing :-

- System testing is a critical phase in the software development life cycle where the entire software system is tested as a whole to ensure that it meets its specified requirements and functions correctly.
- It is conducted after integration testing and before user acceptance testing.

## - 3 Types →

### 1. Functional Testing .

- Functional testing checks the functionality of the software system by testing its features against specified requirements.
- Example :

Unit Testing , Integration Testing .

### 2. Performance Testing .

- Performance testing focuses on the performance like responsiveness, stability, scalability on various conditions to ensure it meets performance requirements.
- Example

Load testing , stress testing

### 3. Non-functional Testing .

- Non-functional testing evaluates aspects of the software system other than its functionality such as Usability, reliability, security.
- Example

Usability , reliability , security testing , testing , testing .

## \* Unit Testing :-

- A unit is a smallest testable part of any software.
- Unit testing is a type of testing where individual unit or components of a software are tested.
- Unit may be a function, method or object.
- It usually has one or a few input and usually one single input.
- It identify and fix bug's at an early stage of development.
- Unit testing ensure that individual parts of the software work properly before integrating them into large modules.

## \* Advantages :-

1. Reduce defects in the newly developed features
2. Reduce cost of testing as defect are capture in early phase.

## \* Disadvantages .

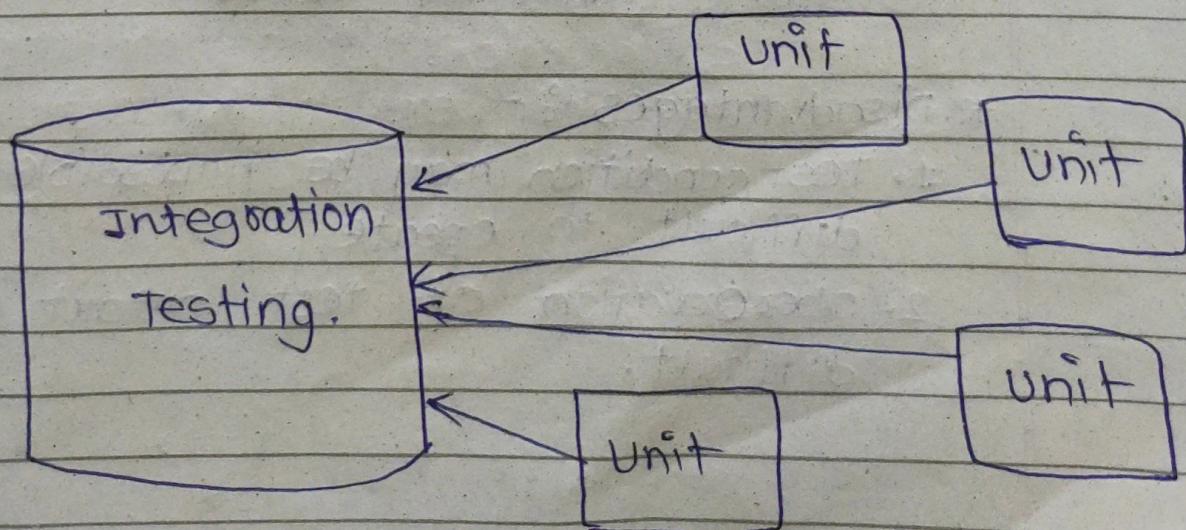
1. Can not catch every expected error.
2. Integrated error's are not solve.

- \* System Test + Integration Testing,
  - Testing that occurs at the lowest level is called unit testing or module testing.
  - Integration testing is the phase in software testing where individual module's are combined and tested as a group.
  - The purpose of Integration testing is finding error's which occur due to combining two module's.
  - It check when two module's are combined then they work properly or not.
  - In this process it work like a incremental testing module in which new or more piece of the software is added and tested until the entire product is test.

#### \* Types of Integration Testing :-

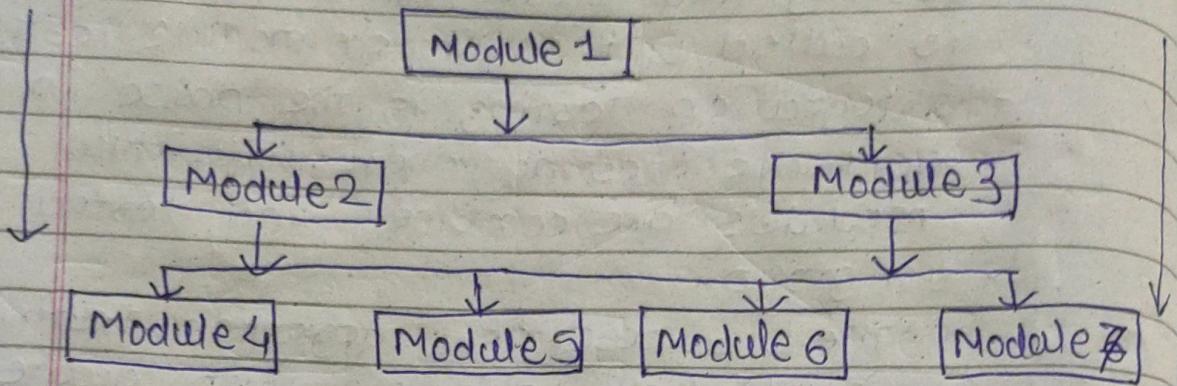
1. Top-down Integration Testing
2. Bottom-up Integration Testing
3. Bidirectional Integration Testing.

#### \* Diagram :-



## \* Types of Integration Testing :-

### 1. Top-down Integration Testing :- (Stub)



- The Top-down integration testing is an incremental approach to integration testing where the highest-level modules or component's are tested first and lower-level modules are integrated and tested step by step.
- In the given diagram the Module 1 is tested first then the Module 2 & Module 3 are tested.

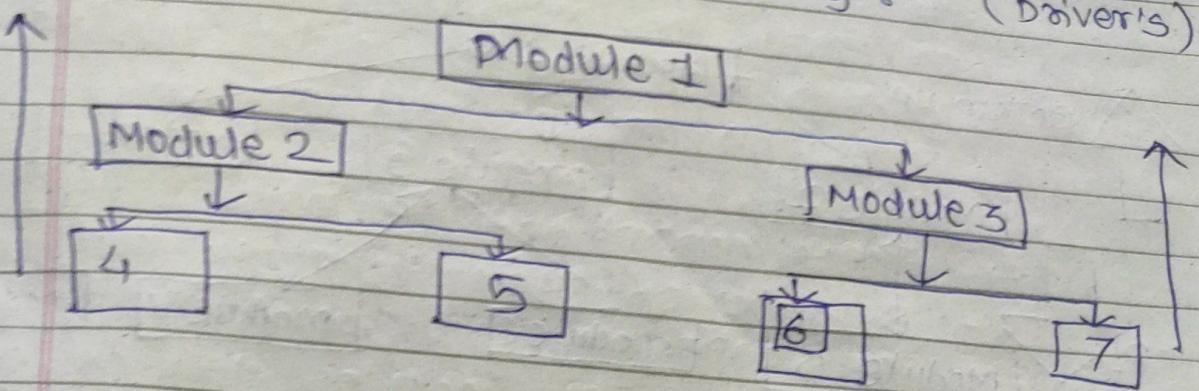
## \* Advantages :-

1. Major problem are cover in top of the program
2. Early identification of high-level design and architectural issues.

## \* Disadvantages :-

1. Test condition may be impossible or very difficult to create.
2. Observation of test output is more difficult.

## 2. Bottom-up Integration Testing :- (Driver's)



- Bottom-up integration testing is an incremental approach to integration testing where testing begins with the lowest-level modules and progresses upward.
- In this approach testing is conducted from sub module to main module.
- In the given diagram the module 4, 5, 6 and 7 tested first then the module 2, 3 and 1 is tested.

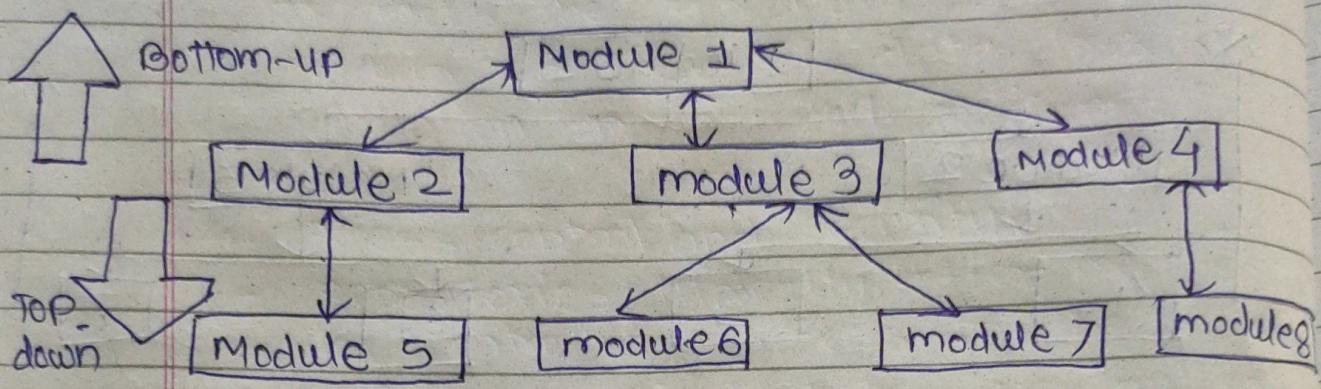
### \* Advantages :-

1. Test condition are easier to create.
2. Observation of test result is easier.

### \* Disadvantages :-

1. Testing done in incomplete testing view.
2. By combining module may cause errors.

### 3. Bidirectional Integration Testing :



- Bidirectional integration is a kind of integration testing process that combines top-down and bottom-up testing.
- It is also called as sandwich testing.
- It is a hybrid approach of testing.
- Bidirectional Approach uses both from top downwards and from the bottom upwards simultaneously.

#### \* Advantages :

1. Useful for very large projects
2. It uses both top-down and bottom-up approach.

#### \* Disadvantages :

1. It is complex.
2. It requires very high cost.

Q 15.

Explain User Acceptance Testing.

### User Acceptance Testing:

- The acceptance testing is a kind of testing conducted to ensure that the software works correctly in the user work environment.
- The Acceptance testing can be conducted over a period of weeks or months.

### Benefits:

#### - Ensure Quality:

Validates that the software is of high quality & ready for release.

#### - User Satisfaction:

Involves user in the testing process, ensuring the final product meets their needs.

#### - Reduces Risk:

Identifies potential issues before the software is deployed, reducing the risk of post-release failures.

#### - Cost-Effective: catching defects early is generally less costly than fixing issues after deployment.

## Types of Acceptance Testing :

1) Alpha Test:

2) Beta Test

1) Alpha Test:

- conducted internally by the customer or employee before the software is released to external users.
- This testing is done under supervision of developer.
- This testing is performed at developer's site.
- 

2) Beta Test:

- conducted by selected groups of customers before final release.
- This testing is done without supervision of developer.
- This testing is performed at customer's site.
- The end user records problem & report them to developer.
- Then developer makes appropriate modification

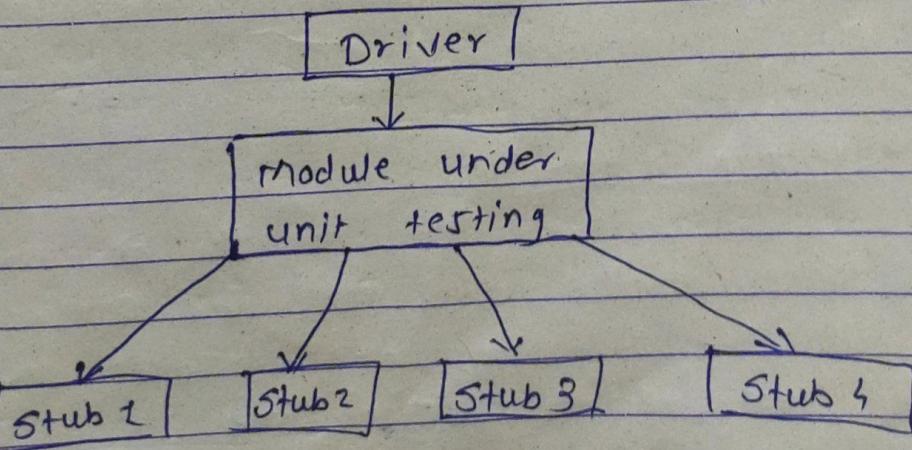
- Q. Explain with suitable diagram drivers & stubs in unit test environment

### Driver :

- These are dummy code.
- These are used in testing when upper level of code is not developed.
- To test lower level code the drivers are used.
- We call the driver as main function which calls other modules to form complete applications.

### Stub :

- These are dummy code
- These are used in testing when lower level of code is not developed.
- To test upper level code the drivers are used.
- It just accepts the value from calling module & returns null value.



- Stub & drivers are two such elements used in testing process which are replacement of module.
- The stubs & drivers are specially developed to meet the necessary requirement of unavailable modules & are useful in getting expected test results.