

## Q.1) What is Testing ?

Ans :- It is a process of executing a program with the intent of finding an error. (or) It is the process of trying to discover every imaginable fault or weakness in work product.

## - Why Testing ?

Primary benefit of testing is that result improved quality, bugs get fixed. Secondary benefit include that software function appear to be working according to specification.

Testing improves usability, Lower's customer support costs (e.g. 24 X 7 Telephone based technical support). Improves reliability, Improves installability, Enhance saleability. (Documentation quality often a selling feature, Recoverability, control and needed adjustment)

## - Why the software has bug ?

- ① Poor Requirement
- ② Mis communication or no-communication
- ③ In experienced programmer and analyst.
- ④ Software complexity and not user friendly
- ⑤ Changing Requirement
- ⑥ User training was poor
- ⑦ Timeline or deadline
- ⑧ Existing hardware proved deficient to handle the new application
- ⑨ Poor documented code.

Q.1) What is bug?

Ans :- A fault in program which causes the program to perform in an unintended or unauthorized manner which is found in development environment.

Q.2) What is error?

Ans :- A mismatch between the program and its specification is an error in the program if and only if the specification itself is correct. It is deviation from actual and expected values.

Q.4) What is defect?

Ans :- An error found in product after shifting it to customer side. An incorrect step process or data definition in computer program within the context of software process is a defect or fault in product.

Q.5) What is Quality?

Ans :- Quality from

(a) A customer view point fit for use or other customer needs

(b) Producer's view point meeting requirement

Quality of software is reasonably bug free, delivered on time, within budget, meets requirement and/or expectations and is "maintainable".

(\*) Verification :-  
Whether the software meets its specification.

Verification is the process of examining the product.

to discover the defects.

Disciplined approach to evaluate whether a software product fulfills the requirements or conditions laid down on them.

"Are we Building the product right?"

Walkthrough, formal inspection and review of code.  
Software product also called as static testing.  
by systematically reviewing the contents of software product with the intention of detecting defects. i.e., in identifying not only presence of defect but also their location.

Verification refers to the set of activities that ensure that software correctly implements a specification.

Functions

## (\*) Validation :-

It is the process of confirming whether the software meets user ~~require~~ requirements with using confirmed integration, system and acceptance testing terms under validation.

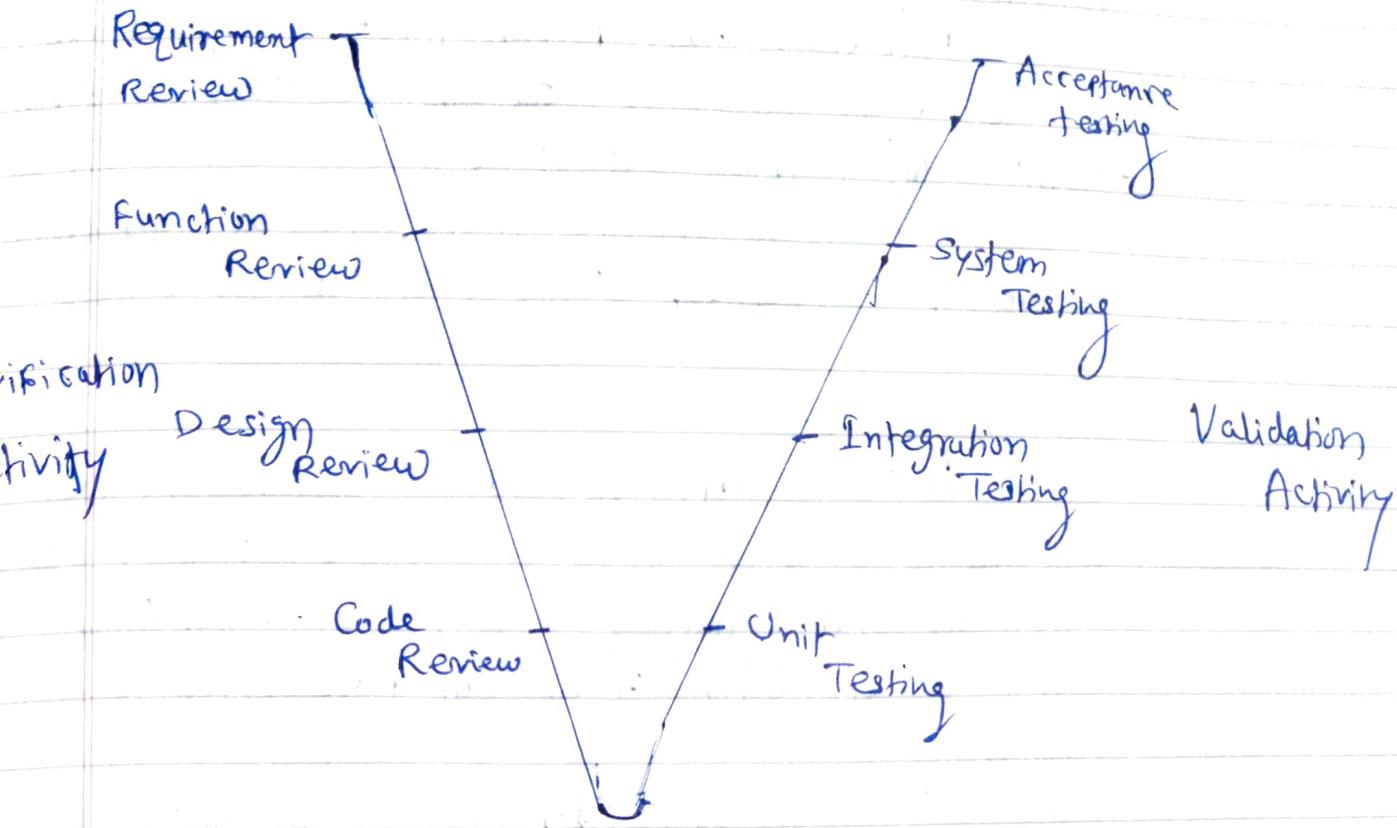
Disciplined approach to evaluate whether the as built software product fulfills its specified intent.

"Are we building the right product?"

Testing each software product at each phase of life cycle using test plan, test cases for properly selected type of test. Also called as Dynamic Testing.

Done by systematically testing a software product with

Intention of finding defects, not their location.



The Activity related to Verification & Validation.

#### 4.2] Software Inspection :-

Inspection are different from Peer review and walkthroughs in that, person who present the code the presenter or reader isn't the original programmer.

This forces someone else to learn and understand the material being presented, potentially giving a different slant and interpretation at the interpretation meeting.

The other participants are called inspectors. Each is tasked with reviewing the code from different perspective, such as user, tester, or product support person.

Some inspectors are also assigned task

Such as moderator and recorder to assure that  
are followed and that the review is run  
efficiently.

Inspections have proven to be very effective  
in finding bugs in any software deliverable such  
as design, documents and code.

They are gaining popularity as company  
and product development teams that discover benefits

Explain

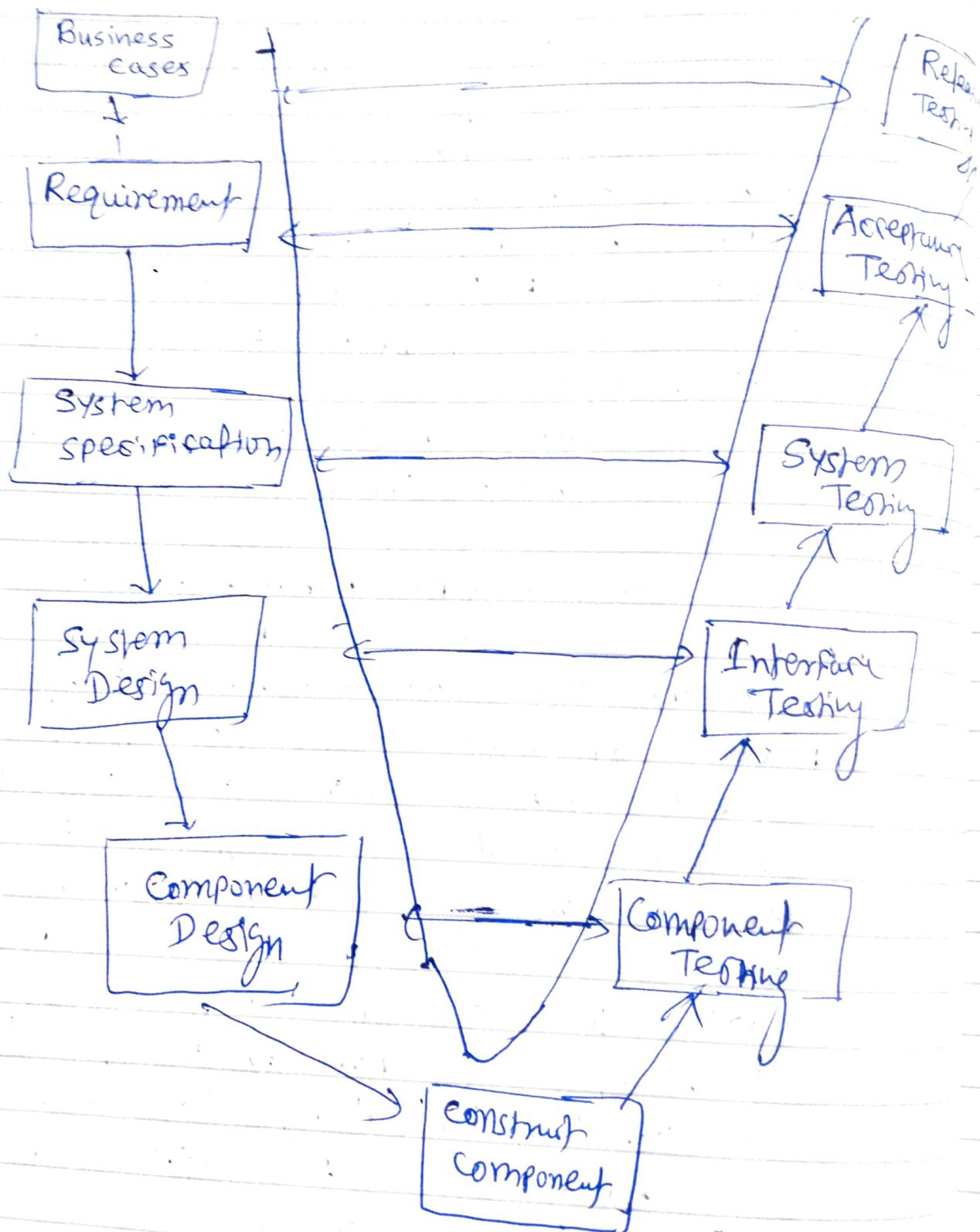


Fig :- The 'V' Model

6.1) Unit Testing :- Unit Testing is a smallest test part of an application. It is nothing but white Box testing to test particular function or code. Typically done by programmer not by tester as it requires detail knowledge of testing, like to exercise program loops called loop testing. ② Testing of individual SW component ③ module.

After completion of design and their reviews programmer concentrate on coding. In that phase programmer conduct program level test, using white box testing technique.

④ Execution Testing :-

- Basis Path coverage  
(All statement in program correctly participated in Execution)
- Loop Coverage  
(All iteration are correctly terminating)
- Program Technique coverage  
(Less no. of memory cycles and CPU cycle during execution)

b) Operation Testing :- To check whether our program run on customer expected platform or not, where Platform means, O.S., compiler, Browser, & other system software.

# Integration Testing

What is Integration Testing?

Testing the features and

The process of combining and testing multiple components together, to ensure that software operates properly when combined together.

This type of testing is especially relevant to client/server and distributed systems. For example, Ball pen testing.

After completion of developed modules developments and testing program compose them to form a system with respect to High level Design.

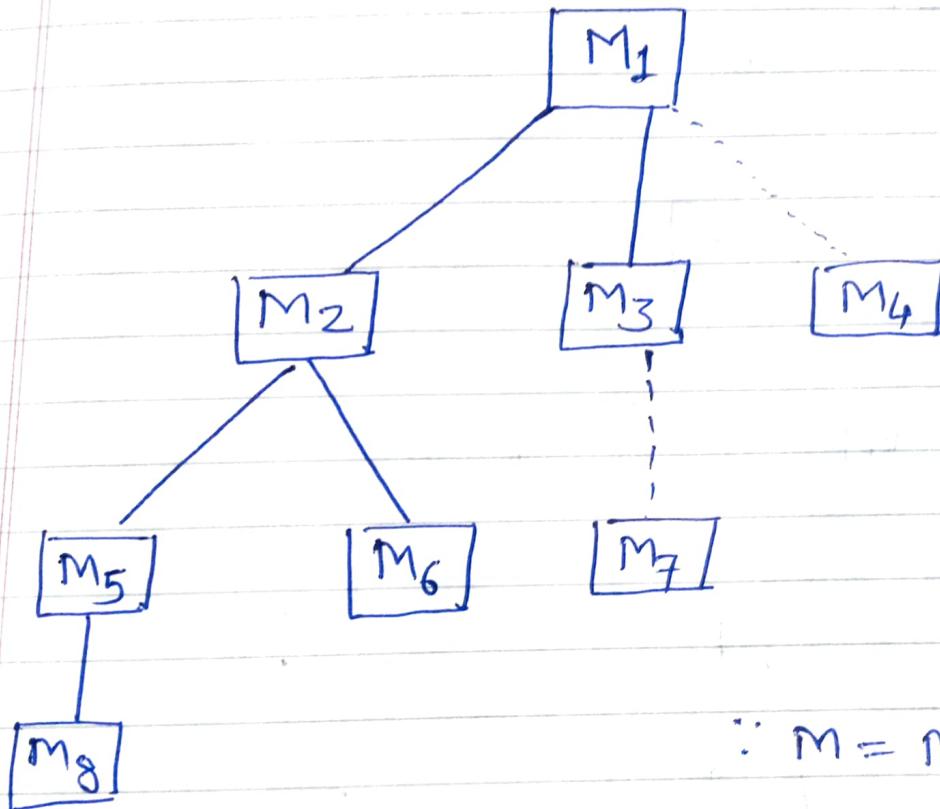
During this they conduct Integration Testing.

It is a systematic technique for constructing the software architecture while at the same time conducting test to uncover errors associated with interfacing.

## Types of Integration Testing :-

### ① Top-Down Integration

Top-Down Integration testing is an incremental approach to construction of software architecture. Modules are integrated by moving downward through the control hierarchy, beginning with main control module (or) Main Program.



② Depth-first Integration - Integrates all components on a major control path of the program structure.

for Example:-

Selecting the left hand Path components

~~Main Program~~

M<sub>1</sub>, M<sub>2</sub>, M<sub>5</sub> would be integrated first

Next M<sub>8</sub> (or) (if necessary for proper functioning of M<sub>2</sub>), M<sub>6</sub> would be integrated.

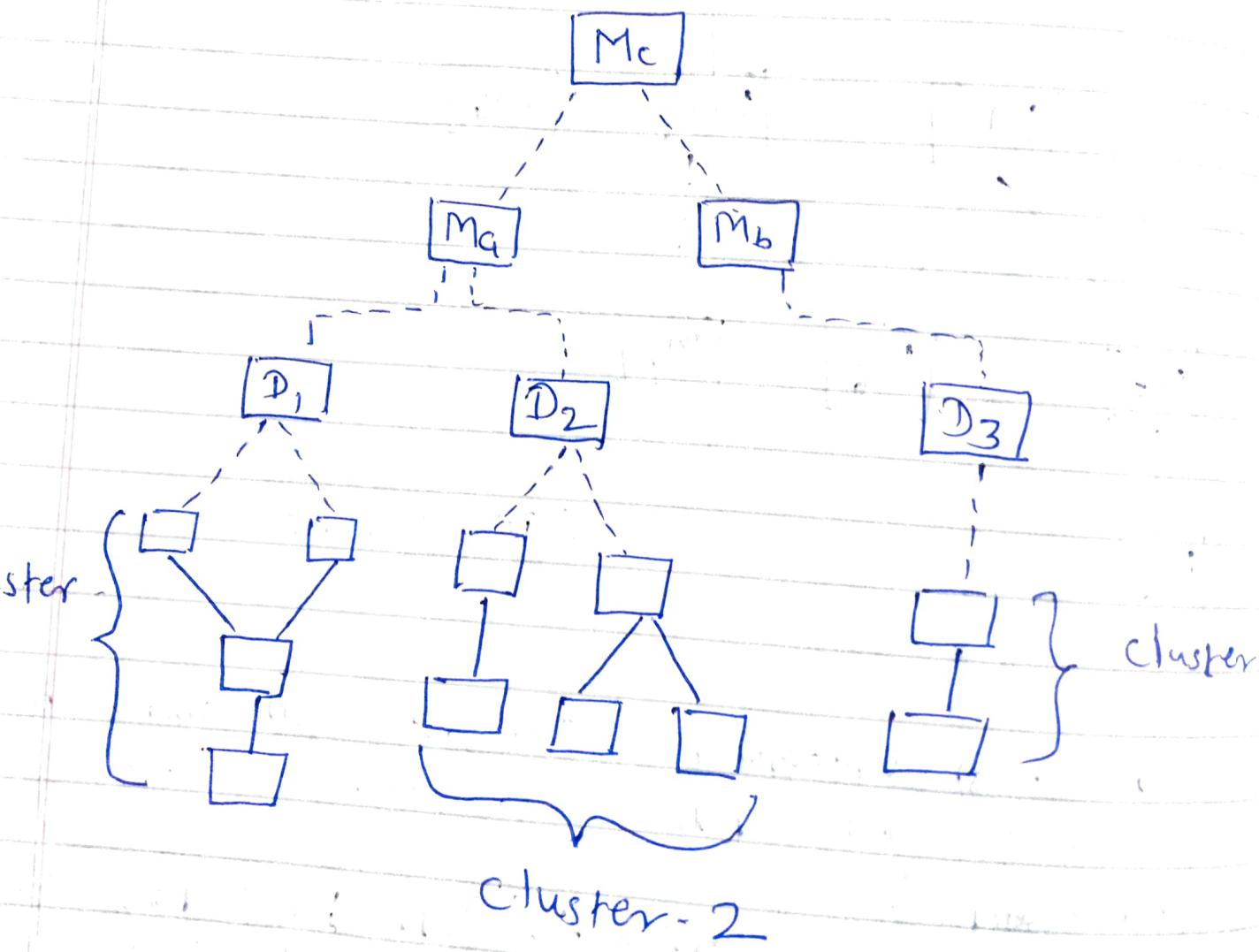
Then the central and right hand

④ Breadth-First Integration - Incorporates all components directly subordinate at each level moving across the structure horizontally.  
From fig.: Component/module  $M_2$ ,  $M_3$  &  $M_4$  would be integrated first. The next level  $M_5$ ,  $M_6$ , and so on - - -

The Top-Down integration strategy verifies major control or decision points early in the test progress.

E)

Bottom-Up Integration



If begins constructing and testing the component at lowest level in the hierarchy.

## 3) System Testing :-

These test simulate operation of entire system and verify that it can correctly. This phase test all the integrated components of a system.

Whole product testing. It is the process of "Validation" to check whether software or system meet its "Verification".

System Testing verifies that all elements mesh properly and that overall system function and performance is achieved.

In System Testing software once validate, must be combined with other system elements e.g. Hardware, people, information, Database

System Testing actually a series of different tests whose primary purpose is to fully exercise the computer based system.

A classic system testing problem is "Finger pointing". This occurs when an error is uncovered and each system element developer blames the other for the problem.

System testing has purpose to verify that system elements have been properly integrated and perform allocated function.

## 4) Acceptance Testing :-

What is Software Acceptance (or) Acceptance Test

Software Acceptance is a incremental process of approving (or) rejecting software systems during development (or) maintenance, according to how the software satisfies predefined criteria.

## 7.1] Installation Testing

To identify the ways in which the installation procedure lead to incorrect results for Installation option like New, upgrade, customized, complete, under normal, Abnormal conditions. It is important as it makes first impression on the end user.

The installation phase is the primary responsibility of the IT department specifically, computer operation personnel have the responsibility for getting the system into operations. The installation process may be stand alone one time process. This enables them to be independent of the development team so that they can perform their installation tasks concurrently with the development process.

Installation function relates not to satisfying user needs but to placing a completed and tested application into production.

The objective of testing is to determine whether the installation is successful therefore the result must be available as quickly as possible. In many instances this means that the test result must be predetermined before the test starts.

## 2] Usability Testing

Testing the ease with which user can learn and use a product so that end user or customer get "User friendly" environment.

for Example :- Learnability, operability, understandability.  
→ Usability is the effort required to learn, operate, prepare input for and interpret output of a program. If the program is not easy to use it may fail. Even if the function that it performs are valuable. Usability is an attempt to quantify ease-of-use and can be measured in terms of characteristics.

Usability evaluates the degree to which user can interact effectively with the application and degree to which application guides users actions, provides meaningful feedback and enforces a consistent interaction approach.

## Usability Categories and objectives :-

Interactivity → Interaction mechanism and easy to understand.

Layout → Navigation mechanism

Readability → Text well written and understandable

Aesthetics → Color, Related characteristics, Feel comfortable for user.

Display → Optimal use of screen, size and resolution.

Accessibility → Accessible to people.

## 73] Regression Testing :-

Retesting of previously tested program following modification to ensure that faults have not been introduced or uncovered as a result of the changes made.

Each time new module is added as a

Part of Integration testing, the software changes. New Data flow paths are established, new Input/Output may occur and new control logic is invoked. These changes may cause problem with functions that previously worked flawlessly.

Regression Testing is the Re-Execution of some subset of test that have already been conducted to ensure that changes have not propagated unintended side effects. Successful test of any kind result in the discovery of errors and errors must be corrected.

Whenever software is corrected, some aspect of the software configuration like program, its documentation or data that support it is changed.

Regression Testing is the activity that helps to ensure that ~~the~~ changes due to testing or for other reasons do not introduce unintended behaviour or additional errors.

## 7.4] Performance Testing :-

Performance Testing conducted to evaluate the compliance (ie. permission or agreement) of a system or component with specific performance

For Ex:- Efficiency throughout, processing speed, Response time, Resource consumption. Performance testing is designed to test the performance of software within the context of

(71)

integrated system. Performance testing occurs throughout all steps in the testing process. Even at the unit level, the performance of an individual module may be assessed as test are conducted.

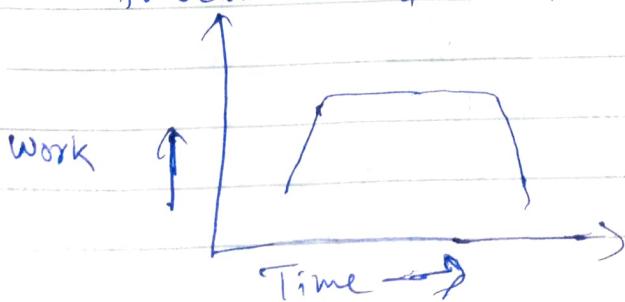
When all system elements are fully integrated that the true performance of system can be ascertained. Performance testing is used to uncover performance problem that can result from lack of server-side resources, inappropriate network bandwidth, inadequate database capabilities, faulty or weak operating system, capabilities poorly designed application, functionality, and other hardware or software issues that can lead to degraded client-server performance. The intend is to do performance testing.

- ① To understand how the system responds to loading (i.e. No. of users, no. of transaction, or overall data volume)
- ② To collect metrics that will lead to design modification to improve performance.

#### 7.4.1] Load Testing :-

Testing an application under heavy load to determine to determine the load, volume, and response time defined by requirement.

For Ex If product has 10 user license software at a time 10 user will login & check the performance to bear a load of software.



Loading is tested at a variety of load levels and its a variety of combinations.

The intent of Load Testing is to determine how the application and its server side environment will respond to various loading conditions.

The Performance  $P = N \times T \times D$  where,

$N$  = No. of concurrent user

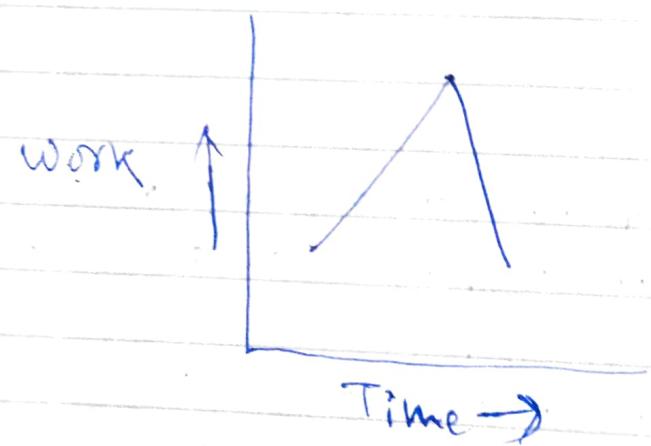
$T$  = The no. of online transaction per user per unit time

$D$  = The Data Load processed by the server per transaction.

#### 7.4.2]. Stress Testing :-

To evaluate a system or component at or beyond the limits of its specified requirement.

for ex. :- If the product has 10 user license software then at a time 10 user will login for prolong period of time. (Reduce the RAM of server)



In Stress testing Loading is increased to the breaking point to determine how much capacity the application environment can handle.

- (15)
- N.T.D. are forced to meet and then exceed operational limits. Stress & testing answer the following questions.
- Does server software generate "Server not available" message?
  - Are transaction lost as capacity is exceeded? If the system does fail, how long will it take to come back on-line?
  - Are certain Application Function discontinued as capacity reaches the 80 or 90% Level...?

### ④ 7.5) Security Testing

- ① Program may be stolen from computer room or storage area
- ② Information Services Facilities may be destroyed or damaged by intruders or employee.
- ③ An un-authorized user may gain access to the system with unauthorized ~~program~~, Password.
- ④ A user may leave a logged in terminal unattended, allowing an unauthorized person to use it.
- ⑤ A terminated employee may retain access to an information services system because his or her name and password are not immediately deleted from authorization tables and control files.
- ⑥ Repeated attempts by the same user or terminal to gain unauthorized access to the system or file may go undetected.
- ⑦ Sensitive data may be carelessly handled by the application staff, by the ~~e-mail service~~ e-mail service or other personnel within the organization
- ⑧ The wrong version of a program may be executed.
- ⑨ A program may be executed twice using same transaction.