

## UNIT-3

### Chapter 07

### Software Testing

#### Topics :-

- (i) Development Testing (Black box testing & white box testing)
- (ii) Test Driven development
- (iii) Release Testing
- (iv) User Testing

#### Development Testing :-

- (i) Unit Testing
- (ii) Choosing unit test cases
- (iii) Component Testing
- (iv) System Testing

#### Test - driven development :-

#### Release Testing :-

- (i) Requirements Based Testing
- (ii) Scenario Testing
- (iii) Performance Testing

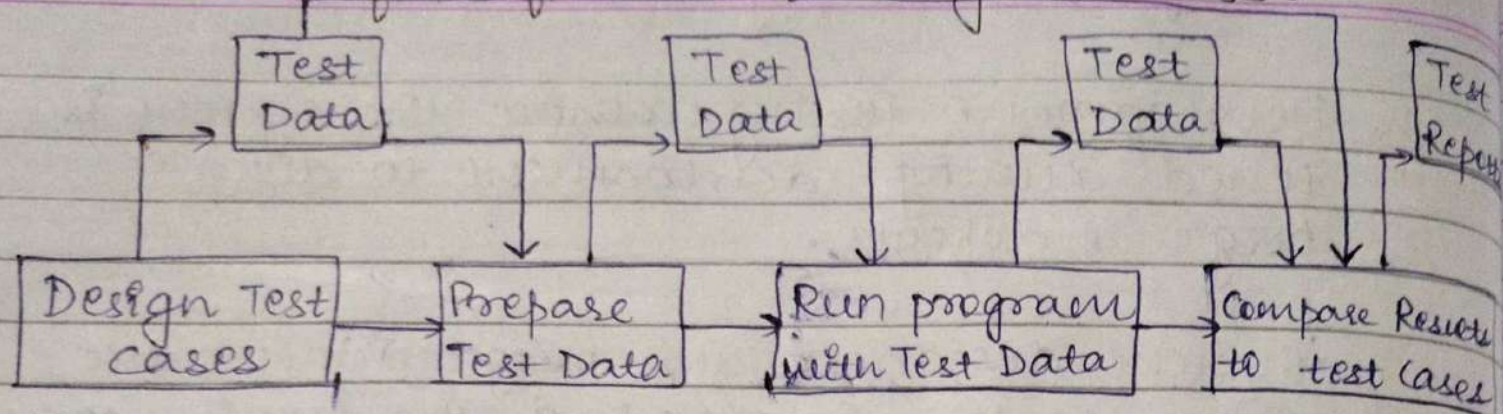
#### User Testing :-

#### \* 03 stages of Testing :-

- 1). Development Testing : where the system is tested during development to discover bugs & defects.
  - System designers and programmers are likely to be involved in the testing process.
- 2). Release Testing : where a separate testing team tests a complete version of the system before it is released to users.
  - The aim of release testing is to check that the system meets the requirements of system and stakeholders.
- 3). User Testing : where users or potential users of a system test the system in their own environment.
  - For software products, the 'user' may be an internal marketing group who decide if the software can be marketed, released & sold.
  - Acceptance Testing is one type of user test where the customer formally tests a system to decide if it should be accepted from the system supplier or if further development is required.



## \* Model of Software Testing Process :-



## \* DEVELOPMENT TESTING :

- It includes all testing activities that are carried out by the team developing the system.
- They are responsible for developing tests and maintaining detailed records of test results.
- During development, testing may be carried out at three levels of granularity :-
  - a). Unit testing: where individual program units or object classes are tested.
    - It should focus on testing the functionality of objects or methods.



b). component testing : where several individual units are integrated to create composite components.  
• It should focus on testing component interfaces.

c). System testing : where some or all of the components in a system are integrated & the system is tested as a whole.  
• It should focus on testing component interactions.

Development Testing is a defect testing process, where the aim of testing is to discover bugs in the software.

\* Black-box testing : When we use the specification of a system to identify equivalence partitions, this is known as black-box testing.

Here, we don't know anything about how the system works.

\* White-box testing : Here we look at the code of the program to find other possible tests.



E.g Exception - Handling to handle a range of incorrect inputs to the code.

\* Testing programs with Sequence, Array or lists :- (Defects) // Possible defects

- 1). If presented with a single-value sequence, a program may not work properly.
- 2). Use diff sequences or different sizes in different tests.
- 3). Derive tests so that the first, middle & last elements of the sequence are accessed. // This approach may reveal problems at position boundaries.



LP

- 4). Given following fragment of code, how many test cases are required for 100% decision coverage?

if width > length  
then biggest dimension = width \*

if ht. > width  
then biggest dim = ht. \*

endif

else  
biggest dim = ~~ht~~ length \*

~~if~~  
if ht > length  
then big-dim = ht. \*

endif  
endif

∴ 4 tests are required for 100% coverage.

- 5). If to the above code, print "Big dim is", & big dim  
print "width:" & width, print "length:" & length.  
How many more test cases required?

further  
Since, no decision checks involved, ~~there~~  
therefore no more test cases required.  
Existing test cases would be sufficient.



6). What is the difference b/w negative & positive testing? Give example for both.

\* **Positive Testing**: It is a type of Software Testing that is performed by assuming everything will be as expected.

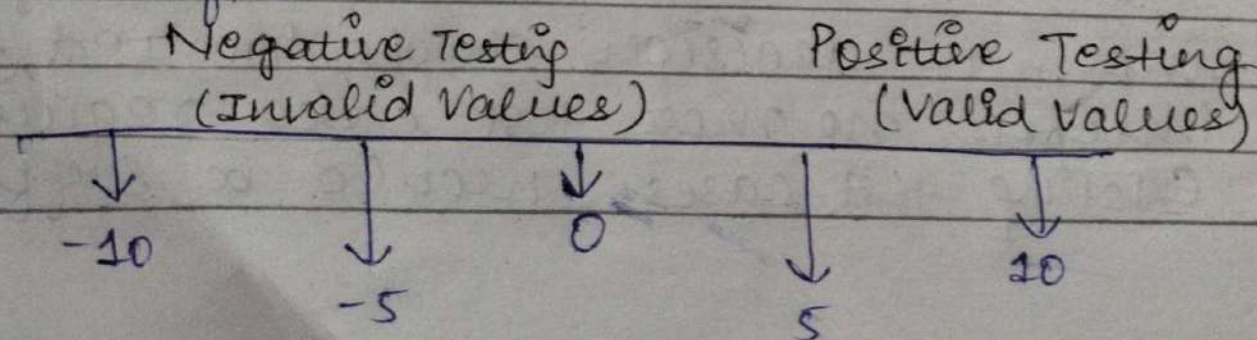
- It is performed with the assumption that only valid & relevant things will occur.

- Data Set & all other functionalities will be as expected.

\* **Negative Testing**: It is a type of software testing that is performed to check the system for unexpected conditions.

- It plays an important role in high-performance software development.

- It checks whether on such unexpected conditions what will be the behavior of the software.





## Positive Testing

It is performed only for expected conditions.

It doesn't cover all possible cases.

It doesn't ensure a good quality product.

It can be performed by people having less knowledge.

It takes less time.

It is performed on every application.

It ensures software is normal.

## Negative Testing

It is performed for unexpected conditions.

It covers all possible cases.

It ensures a good quality product.

It can only be performed by professionals.

It takes more time.

It is performed where are chances of unexpected conditions.

It ensures 100% defect free software.



7). You have written module to replace old string with new string if there is a match. Write test cases to perform the same.

(i) If the 'match' is presented with a single-value sequence, then the program may not function correctly.

(ii) ~~If the size~~ The size of the old string & the new string should be checked.

(iii) There might be a need to check the old string at the partitions to compare with the match.

(iv) If the size of 'match' is greater than the string length then it might cause buffer overflow.

(v) If the old string & the new string are same, then the program may ~~result~~ run numerous times (infinite loop).

(vi) If the size of the old or new string is greater than the string length then invalid outputs can be generated.



1) What are the different levels of testing and the goals of different levels? For each level, specify which of the testing approaches is most suitable.

There can be many ways to differentiate the different levels of testing.

A/q to the basic model of software testing process, there are 03 levels of testing & the approaches are:-

- ① Development Testing: Unit/Component testing, System Testing
- ② Release Testing: Requirements Based testing, Scenario testing, Performance testing.
- ③ User Testing: Acceptance testing.

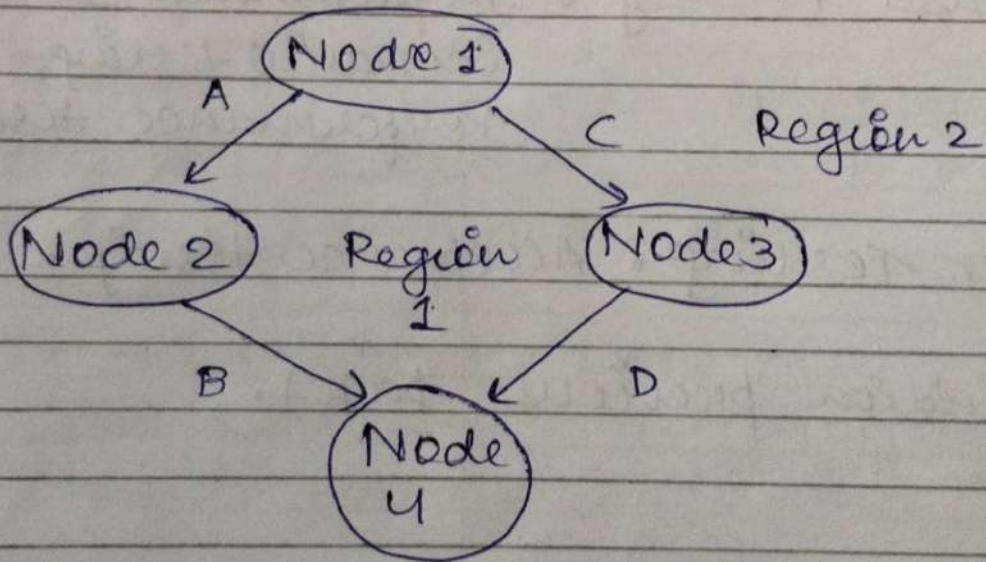
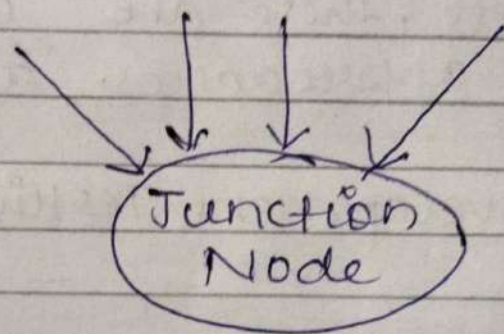
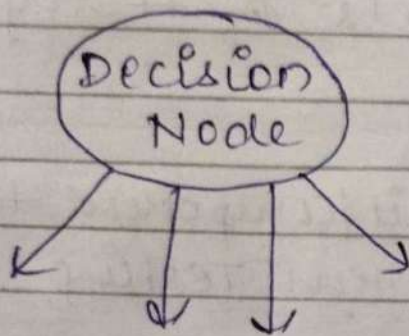
(definition previous page).



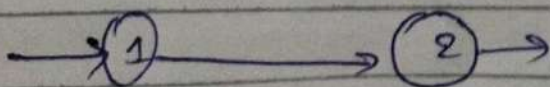
# ★ PATH TESTING :-

- 1). Construct the Control Flow Graph.
- 2). Compute the cyclomatic complexity of the graph.
- 3). Identify the Independent paths.
- 4). Design test cases from independent paths.

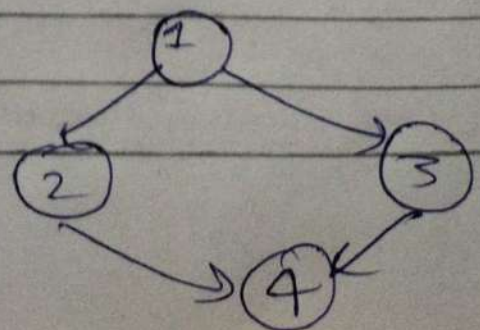
## 1). Control Flow Graph :-



• Sequence

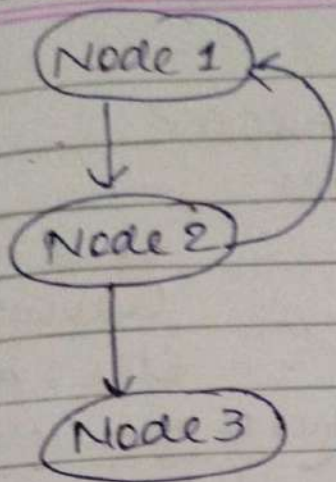


then  
• If  $\neg$  Else

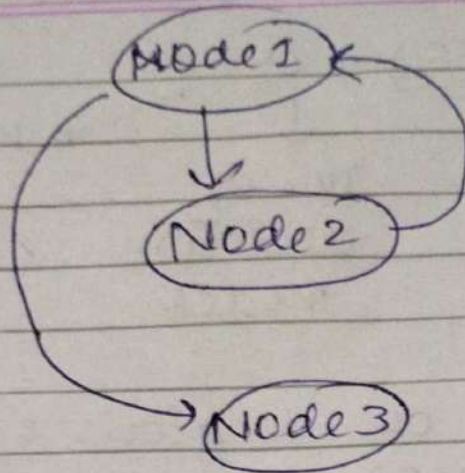




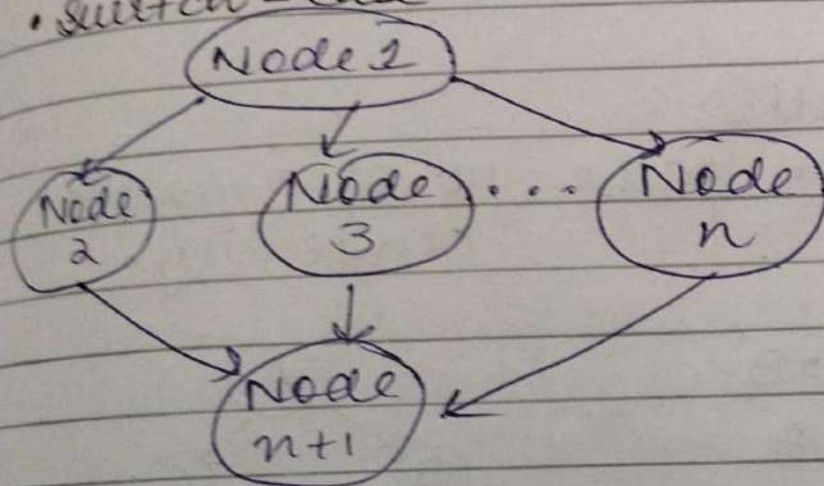
• do-while



• while-do



• Switch-case



2) Cyclomatic Complexity :-

// Based on Edge & nodes

$$V(G) = e - n + 2 * p$$

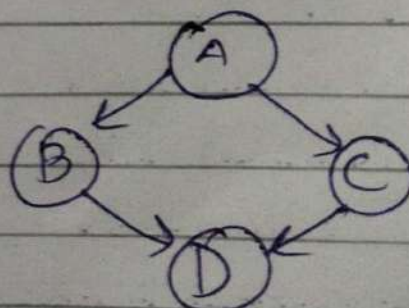
$e$  = no. of edge

$n$  = no. of vertices

$p$  = no. of Connected Components

E.g  $e = 4$   $p = 1$   
 $n = 4$

$$\therefore V(G) = \underline{2}$$





// Based on decision nodes :-

$$V(G) = d + P$$

↓  
no. of  
decision  
nodes

→ no. of connected  
nodes

If 2 arrows  
leaving:  $d = 1$   
else:  $d = K - 1$

no. of arrow leaving

// Based on regions :-

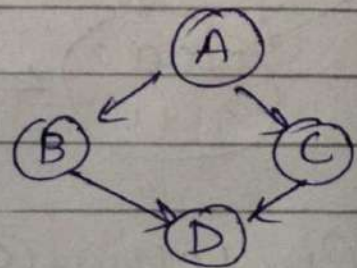
$V(G) = \text{no. of regions in the graph.}$

3). Independent Paths :

no. of independent paths = no. of cyclomatic complexity.

∴ Path 1 :  $A \rightarrow B$

Path 2 :  $C \rightarrow D$

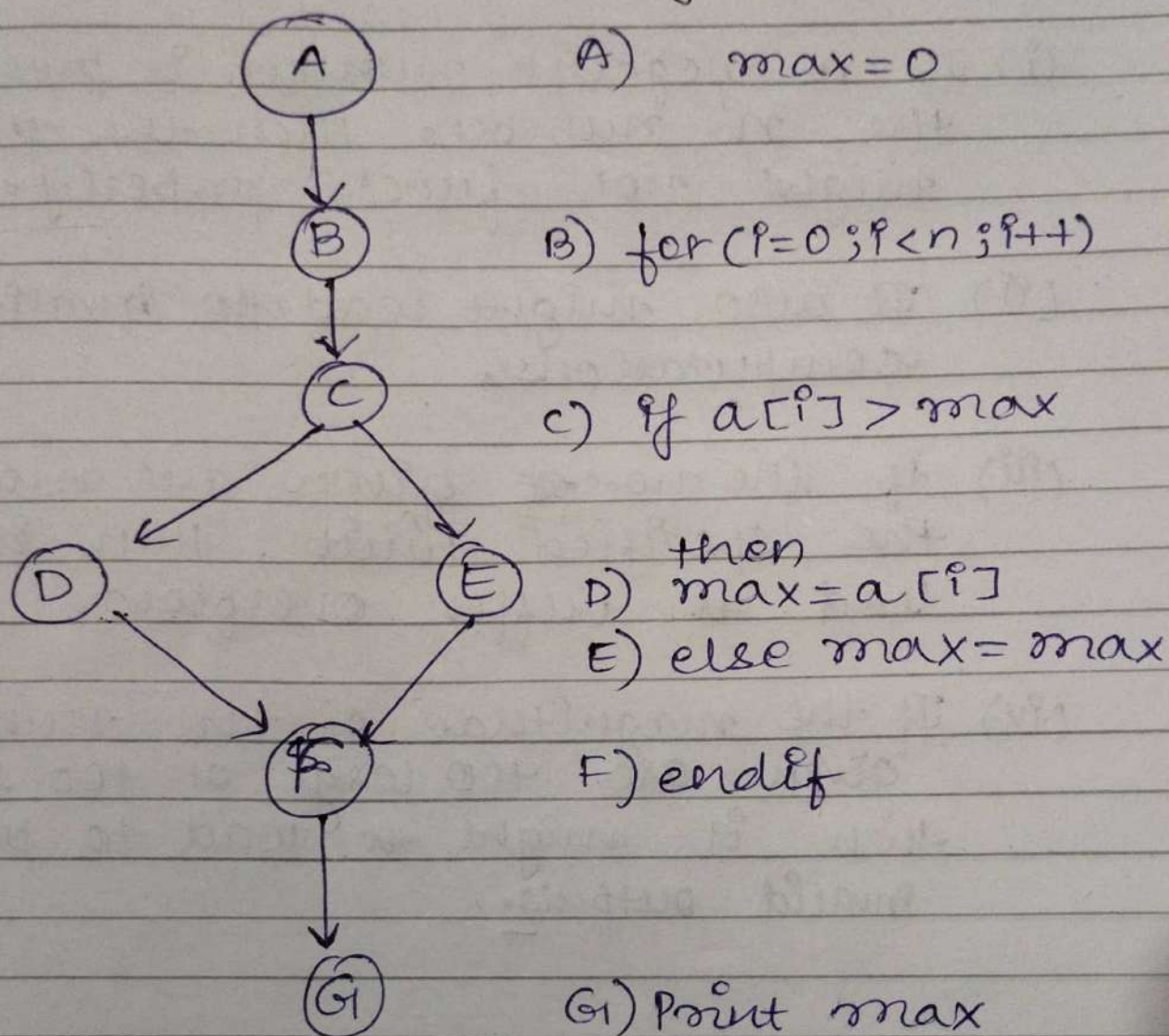


4). Design Test Cases :

Test cases can be designed where each test case represents one or more independent paths.



9). You need to find largest of N nos. Design test cases using path testing technique.



Here,  $V(G) = C + P$   
 $= 1 + 1 = \underline{\underline{2}}$

So, two independent paths are there.

Path 1: ~~A → B → C → D → F → G~~ A → B → C → D → F → G

Path 2: ~~A → B → C → E → F → G~~ A → B → C → E → F → G



Therefore, test cases would be :-

- (i) If a negative number is present in the 'n' numbers then the program might not function properly.
- (ii) It also might lead to invalid output computations.
- (iii) If the no.s of given are more than the specified limit, then it might lead to buffer overflow.
- (iv) If the magnitude of the numbers given are too large or too small, then it might lead to produce invalid outputs.



## Chapter 07

### Software

### Testing

#### // Sample Questions

6). Explain following terms:-

a). VERIFICATION & VALIDATION:-

Validation: "Are we building the right product?"

Verification: "Are we building the product right?"

- Verification & Validation processes are concerned with checking that software being developed meets its specification & delivers the functionality expected by the client.
- The aim of verification is to check that the software meets its state functional & non-functional requirements.
- The aim of validation is to ensure that the software meets the customer's expectations.



- Validation is essential because requirements specification do not always reflect the real wishes or needs of system users.
- The ultimate goal of verification & validation process is to establish confidence that the software system is ~~'#'~~ good enough for its ~~int~~ intended use. | 'fit for purpose'.

## b). UNIT TESTING:

- Unit Testing is the process of testing individual components in isolation.
- It is a defect testing process.
- Units may be :-
  - Individual functions or methods within an object.
  - Object classes with several attributes & methods.
  - Composite components with defined interfaces used to access their functionality.



### c). SYSTEM TESTING:

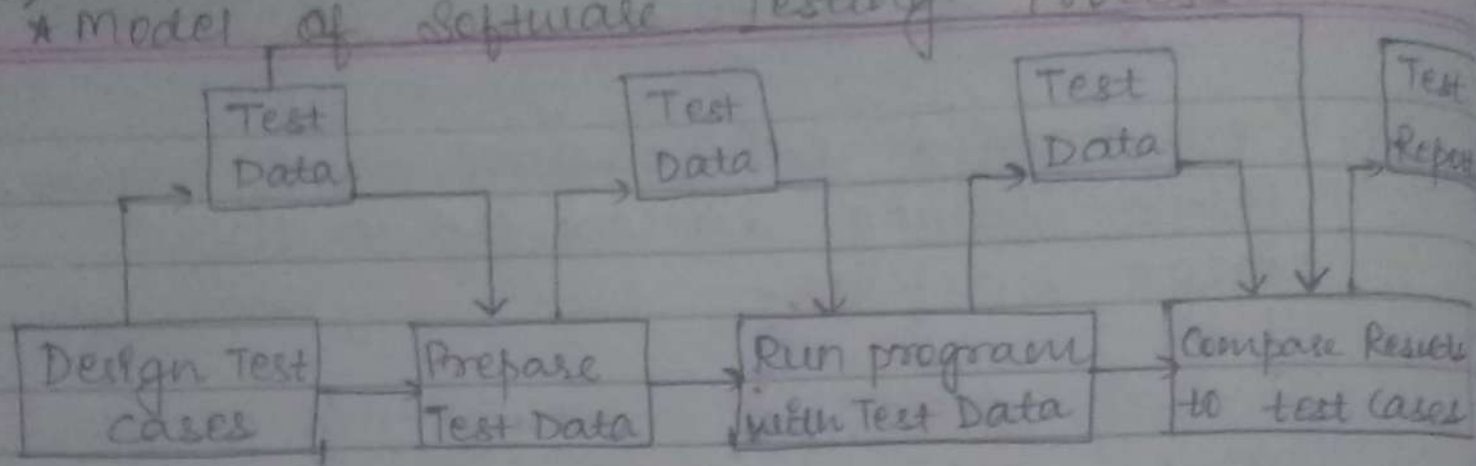
- It involves integrating components to create a version of the system & then testing the integrated system.
- The focus in system testing is testing the interactions b/w components.
- It checks that components are compatible, interact correctly & transfer the right data at the right time across their interfaces.
- It tests the emergent behaviour of a system.

### d). PERFORMANCE TESTING:

- Tests should reflect the profile of use of the system.
- It usually involve planning a series of tests where the load is steadily increased until the system performance becomes unacceptable.
- Stress testing is a form of performance testing where the system is deliberately overloaded to test its failure behaviour.



③: A Model of Software Testing Process:-





### \* 03 stages of Testing :-

- 1). **Development Testing**: where the system is tested during development to discover bugs & defects.
  - System designers and programmers are likely to be involved in the testing process.
- 2). **Release Testing**: where a separate testing team tests a complete version of the system before it is released to users.
  - The aim of release testing is to check that the system meets the requirements of system and stakeholders.
- 3). **User Testing**: where users or potential users of a system test the system in their own environment.
  - For software products, the 'user' may be an internal marketing group who decide if the software can be marketed, released & sold.
  - Acceptance Testing is one type of user test where the customer formally tests a system to decide if it should be accepted from the system supplier or if further development is required.