

## Software testing

- objective -
- gain customer confidence
  - find defects
  - gain confidence by providing quality
  - prevent defect
  - end result meet requirement

- **Fault** - A static defect in the software. It could be a missing function or a wrong function in code.
- condition that cause software to fail to perform required function.
- **Failure** - An external, incorrect behaviour with respect to the requirement. A failure is manifestation of fault when software is executed. - inability of a system or component to perform required function according to specification.
- **Error** - An incorrect internal state that is manifestation of some fault. difference between actual and expected output.

Error is term used by developer.

Bug is term used by tester. It is informal name of defects.

Software testing - Software testing can be stated as the process of verifying and validating whether software or application is bug free, meets technical requirement or guided by design & development and user requirement effectively and efficiently by handling all boundary.

- **Verification** - It refers to set of task that ensure that software correctly implements a specific function.
- **Validation** - It refers to different set of task that ensure software that has been built is upto customer requirement.

## Types of software testing

\ /

manual testing      Automation testing.

- \* **Manual testing** - Manual testing includes testing software manually without using tool or any script. In this tester takes over role of end user and test software to identify unexpected behaviour or bug.

There are different stages for manual testing such as unit testing, integration testing, system testing, user acceptance testing.

Testers use test plan, test case to test software.

- \* **Automation testing** - In this tester write scripts and uses another software to test the product. Automation testing is used to re run test scenarios quickly and repeatedly that were performed manually in manual testing.

It test application from a load, performance and other point of view. It increases test coverage, improve accuracy and save time and money.

different types of software testing techniques -

- \* **Black box testing** - The technique of testing in which the tester doesn't have access to the source code of software and it conducted at the software interface with concern with internal logical structure.

- \* **White box testing** - technique of testing in which tester aware of the internal working of product has

access to source code and it conducted by making sure that all internal operation are performed according to specification.

### Black box testing

- way of software testing in which internal structure of program code is hidden.
- implementation of code not needed.
- mostly done by software tester.
- It can be referred to as outer or external software testing.
- It is a functional test of software.
- No knowledge of programming.
- It is behaviour testing.
- also called closed testing.
- Types - functional testing  
non-functional testing  
Regression testing.
- less exhaustive task.
- less time consuming.

### White box testing

- way of testing software the tester has knowledge about internal structure of code.
- Code implementation is necessary.
- It is mostly done by software developer.
- It is the inner or internal testing.
- It is structural test of software.
- Mandatory knowledge of programming.
- It is logic testing.
- also called clear box testing.
- Types - Path testing  
loop testing.  
condition testing.
- more exhaustive task
- more time consuming.

### \* Software testing principles -

for testing application, we need to follow 10m principle.  
there are -

- Testing shows the presence of defects - (testing help find defects)
- Exhaustive testing is impossible (testing every module is not easy to test important ones)
- Early testing - testing should start at early stage.
- defect clustering - it means throughout process no. of bugs are correlated to smaller modules. but this method do not identify new defects.
- Peppermint paradox - same test will not find new bugs. to get over from this review all test case.

- testing is context dependent - we have different kinds of techniques and approach to solve different software. Ex- e-commerce & commercial site are tested differently.
- Absence of error fallacy - once application is completed tested and there are no bug so we can say 99%. app is bug free. absence of error fallacy means identifying and fixing bug would not help if app is impractical to client need.

### Testing

- Process of finding bug and error debugging is process of to correct bug found during testing.
- It is process to identify failure of implemented code.
- Done by tester.
- can be done by insides or outside.
- manual or automated
- Testing is 'initiated' after code is written.

Bug tracking - It is a software application that keeps track of reported software bugs in software development projects.

- Objective - track so that management may not miss any defect from correction
- saves time
  - help fast delivery of project.

Static testing - type of software testing method which is performed to check defects in software without actually executing the code.

- Performed in early stages.
- Easily be fixed.

### Static testing

#### Review

- informal
- walkthrough
- Peer review
- inspection

#### Static Analysis

- Data flow
- control flow
- cyclomatic complexity

Static analysis is done using various tools, or we can say developed code is analyzed with some tool for structural bugs which might cause defects.

It help identify -  
 dead code  
 unused variable  
 endless loop  
 incorrect syntax  
 variable with undefined value .

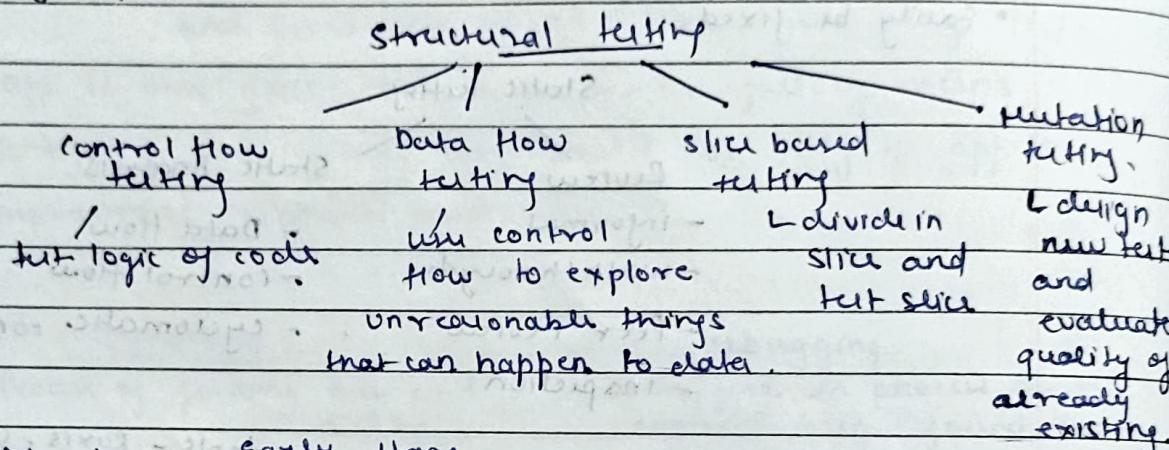
Data flow - connected to stream testing.

control flow - how command and instruction are implemented.

cyclomatic complexity - measurement of program complexity linked to number of independent paths in control flow graph of program.

Tools static testing - check style, source meter, sonar.

structural testing - It is basically related to the internal design and implementation of software. It is performed by team who knows development phase of software.



Advantage - Early stage  
- eliminate dead code  
- automated

Disadvantage - code knowledge required  
- training required  
- expensive

Tool - JBehave, JUnit

- code complexity testing - software metric used to indicate complexity of program. - computed using control flow graph.

If no control statement then it is 1. If it contain if condition then it is 2.

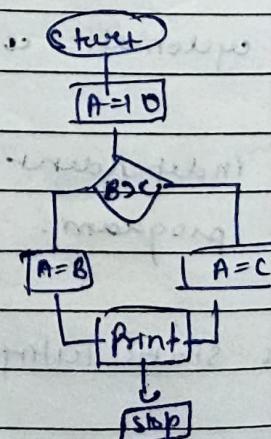
$$M = E - N + 2P$$

E = No of edges.

N = No. of nodes:

P = connected component:

$$\begin{aligned} M &= 7 - 7 + 2 \\ &= 2 \end{aligned}$$



Advantage - easy to apply

- quality metrics.

disadvantage - measure program control & not data complexity.

- harder to understand.

Unit testing - it is a software testing technique by means of which individual unit of software i.e. modules are tested to determine whether they are working or not.

objective - isolate section of code

verify correctness of code.

test every function

help reduce code reuse.

tools - Junit, Jtest.

Advantage - functionality of each unit is known to tester.

- refine code and

disadvantage - time consuming

not efficient.

- code coverage testing - software testing metric which determine how much code is tested. which help in assessing quality of test suite and analyze how comprehensive software is.

criteria - statement coverage.

decision coverage.

function coverage.

condition coverage.

Tool - cobertura, clover

Adv - define performance & quality  
easy maintenance of code

efficiency of implementation

disadv - fail to cover whole code

fails now perfectly  
code has been covered.

Requirement based testing - Requirement based testing is testing approach in which test case, conditions and data are derived from requirement. It includes functional tests.

Stages - design test case

execute test

verify test result

verify test coverage.

track and manage.

\* Boundary value analysis - Based on testing the boundary values of valid and invalid partitions. The behaviour at edges more likely to be incorrect. Check 'input' near boundary.

Ex - consider a system that accept age 18 to 56

valid test case can be - 18, 19, 55, 56

invalid case = 17, 57

\* Equivalence Partitioning method - Divide input domain into classes of data and with. Test cases can be derived.

Example - consider any college admission process based on percentage.

Accept - 50 to 90%.

more or less not accepted

equivalence partitioning.

invalid	Valid	invalid
---------	-------	---------

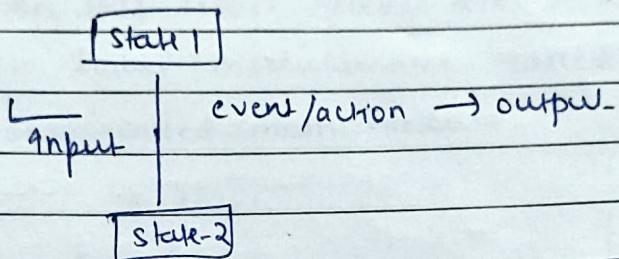
$\leq 50$

$50-90$

$\geq 90$

State transition testing - software testing which is performed to check the change in state of application under varying input. the condition of input passed is changed and change in state is observed.

kind of black box testing.



- objective → test behaviours of system
- test change in transition
  - test performance.

Adv - understand behaviour  
- cover all condition

disadv - cannot be performed anywhere  
- not reliable.

Model based testing - it is nothing but simple testing on which we get different test case that described by model. test case are generated both online and offline test case model.

Importance - Unit testing is not sufficient to check functionality.

- ensure system is behaving in same sequence of action.
- commercial tools are available in market.

Adv

- high level automation
- change in model reflect in different tests.
- Exhaustive testing is possible

disadv -

- require model to carry out testing
- change in model change test.
- test case are highly coupled to model.