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 funⁿ.

• 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 funⁿ 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 as 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 stress 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 is conducted at the software interface with concern with internal logical structure.

- white box testing - technique of testing in which tester is aware of the internal working of product has access to source code and it's conducted by making sure that all internal operation are performed according to specification.

Black box testing	white box testing
way of software testing in which internal structure or program code is hidden	way of testing software the tester has knowledge about internal structure or code.
Implementation of code not needed	code implementation is necessary.
mostly done by software tester.	It is mostly done by software
It can be referred to as outer or external software testing.	It is the inner or internal testing.
It is a functional test of software	It is structural test of software.
No knowledge of programming	Mandatory knowledge of programming.
It is behaviour testing	It is logic testing.
also called closed testing.	also called clear box testing.
Types - functional testing Non-functional testing Regression testing.	Types - Path testing Loop testing. condition testing.
less exhaustive task.	more exhaustive task
less time consuming	more time consuming.

* software testing principles -

for testing application we need to follow some principles. these are -

- Testing shows the presence of defects. (testing help find defects)
- Exhaustive testing is impossible (testing every module is not easy so test important ones)
- Early testing - testing should start at early stage.
- 80/20 rule - defect clustering - it means throughout process no. of bugs are correlated to smaller modules. but this method do not identify new defects.
- Pesticide paradox - same test will not find new bugs. to get over from this review all testcase.

- testing is context dependent - we have different kinds ; technique 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	debugging
- 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.	It is process to give absolution to code.
- Done by tester.	done by programmer or developer.
- can be done by insider or outsider.	done by insider.
- manual or automated	manual.
- Testing is initiated after code is written.	debugging commences with execution of test case.

Defect 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 diffect 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 stage.
- Easily be fixed.

Static testing

Review
- informal
- walkthrough
- Peer review
- inspiction

Static Analysis
- Data flow
- control flow
- cyclomatic complexity.

tools - Raxis, SonarQube

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 undefine 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 - checkstyle, sourcemeter, Soot.

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.

Structural testing

- control flow testing
  - test logic of code.
- Data flow testing
  - uses control flow to explore unreasonable things that can happen to data.
- slice based testing
  - ↳ divide in slice and test slice
- mutation testing.
  - ↳ design new test and evaluate quality of already existing.

Advantage - Early stage
- ~~eliminate dead code~~
- automated.

disadvantage - code knowledge.
training is required
expensive

tool - J behave, 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.

$M = E - N + 2P$

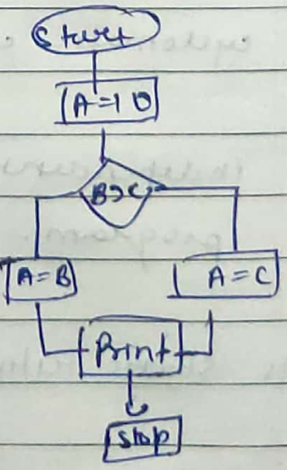
E = No of edges.

N = No. of nodes.

P = connected component.

$M = 7 - 7 + 2$

$= \underline{\underline{2}}$



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 how 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. Include functional tests.

stage - design test case
Execute test
verify test result
verify test coverage.
track and manage.

* Boundary value analysis - Based on testing the boundary value of valid and invalid partitions. the behaviour at edge 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 help of these classes of data. test can be derived.

Example - consider ~~exa~~ any college admission process based on percentage.

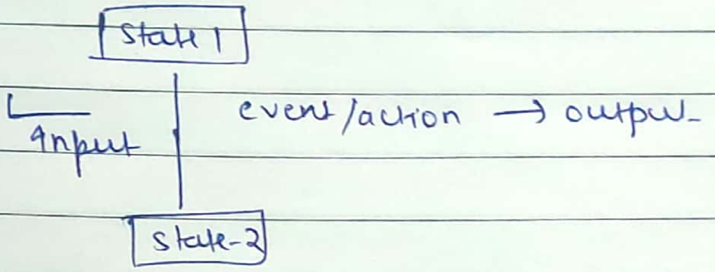
Accept - 50 to 90%.
more or less not accepted

equivalence Partitioning.

invalid	valid	invalid
<=50	50-90	>=90

state transition testing - software testing which is performed to check the change is 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 behaviour of system
- test change in transition
- test performance.

Adv - understand behaviour
- covers all condition

disadv - cannot be performed anywhere
- not reliable.

Model based testing - It is nothing but simple testing in 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 result 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.