

## Software Testing

Q1 ~~Q1~~ alpha, beta, unit, Integration, System & Regression Testing

Sol ~~Q1~~ Alpha Testing → Alpha testing is a type of software testing conducted by the internal development team before releasing the software to external users. It is performed to identify bugs. It focuses on testing the software in a controlled environment to identify any defect early in the development process.

Beta Testing → It is done by real users of the software application in a real environment. A pre-release version of the product is made available for testing to a chosen set of external users during second phase of software testing. It helps to identify any remaining issues for improvement before the software is released in public.

Unit Testing → It is a process of checking small piece of code to ensure that the individual part of a program work properly on their own.

- It is used to test individual ~~test~~ block of functionality.
- It is done by developers.
- Focus on smallest element of software i.e. a module.

## Integration Testing →

- Integration Testing is the process of testing the interaction b/w different components of software.
- Integration testing helps to identify any issue that may arise due to interaction b/w components.
- It occurs after unit testing & before system testing.
- types - Big Bang, Bind, Top down, bottom up.

## System Testing

- It is a process of testing the entire software as a whole.
- It is a part of system test to evaluate the correctness of system test.
- It covers functional & non-functional aspects of software.
- Black box testing

## 6. Regression Testing -

- It is the process of retesting previously tested software after changes have been made to ensure that existing functionalities are not affected
- It helps to maintain the stability & reliability of the software over time as it evolves.

### Types -

- full Regression Testing → Testing the entire application from start to finish after changes have been made
- Partial RT → Testing only those parts of the application that were affected by the changes.

Q2

## White Box Testing → Software Dev

- It is also known as clear box testing, glass box testing, transparent box.
- It is a method of software testing that test internal structure or working of an application.
- White box testing focuses on understanding & testing the internal logic, paths & structure of the code.
- top white box testing tools - Nutt, Veracode, CppUnit

## Applications →

- It ensures that every part of the code is checked including different conditions.
- It makes sure that the flow of action in the code like loops & if statement works properly.
- It finds & test all the possible ways the code can run.
- It checks how different parts of the software work together to make sure they connect smoothly.

## Challenges —

- It can be complex especially for large software applications with extensive databases.
- It may require significant time & effort to design & execute white box tests.
- As software evolves & changes over time, maintaining & updating white box tests to reflect these changes can be challenging.

## Merits →

1. It checks all parts of the software inside out, ensuring nothing is missed.
2. It finds problems early, which saves time.
3. It organizes code, making the software better.
4. It helps make the code faster & more efficient.

## Demerits →

1. It needs to know how the code is written, which can lead to missing some issues.
2. It can be tough to check every possible way the code can run, leaving some parts untested.
3. It might not catch problems with how ~~does~~ users interact with the software with external parts.
4. It takes time to maintain.

Q 3

## Black Box Testing - software tester

It is a software testing technique where the internal working of the system under test are not known to the tester. The tester interact with software solely through its external interfaces, providing inputs & output to assess its behaviour.

### Afflications →

1. It is commonly used for functional testing.
2. It is useful for regression testing to ensure that new updates haven't affected existing functionality.
3. Testing the compatibility of software with different OS, browser or device.
4. It can find security issues by looking at the software from outside without need to know how it's programmed internally.

### Challenges:-

- It is hard to test everything because tester can't see the code.
- finding out why something went wrong can be tough.
- Testing get trickier when the software is really complicated & tester doesn't know how it all fits together inside.

## Merits →

- It is efficient for implementing the tests in the larger system
- Tests are executed from the user point of view
- It can also test across different browsers & OS
- Black box testing is unbiased & best for testing
- It can be initiated at an early stage of the development of software.

## Demerits →

- Without clear functional specification test cases are difficult to implement
- It does not provide a detailed explanation.
- It provides with outer knowledge . black box testing can be used with other testing approach like white box & grey box testing
- Some programs in the application are not tested .

## ~~Q~~ Explain Debugging with its tools.

~~Q~~

Debugging is the process of identifying & fixing errors or bug in software to ensure its functions correctly.

It involves examining the code, analyzing its behaviour & making adjustment to correct any issues.

Debugging can be time-consuming & complex task, but it is essential for ensuring that a software system is functioning correctly.

Tools →

1. IDE → IDEs like V.S, IntelliJ Idea come with built-in debugging features. They offer tools such as break point etc.

2. Radare 2 → It is known for Reverse engineering framework. It is made up of a small set of utilities. It is known as r2.

3. WinDbg → It is a debugging tool designed for Microsoft Windows OS. This tool can be used to debug the memory dumps.

4. Valgrind - It is one of its popular tools which can successfully detect memory related errors caused in C & C++ programs as it may crash the program & result in unpredictable behaviour.

5. firebug - A browser extension for Mozilla now integrated into Firefox Developer Tools, offering debugging capabilities for HTML, CSS & JS.

~~Q \$~~ Cyclomatic Complexity or Macabe's Path Method with example

~~John~~ Cyclomatic Complexity also known as Cyclomatic Complexity or Macabe's Path Method. It is Software Metric used to Measure the complexity of a program. It quantifies the no. of linearly independent path through a program's source code, representing the no. of possible execution paths.

It can be determined by any three formula

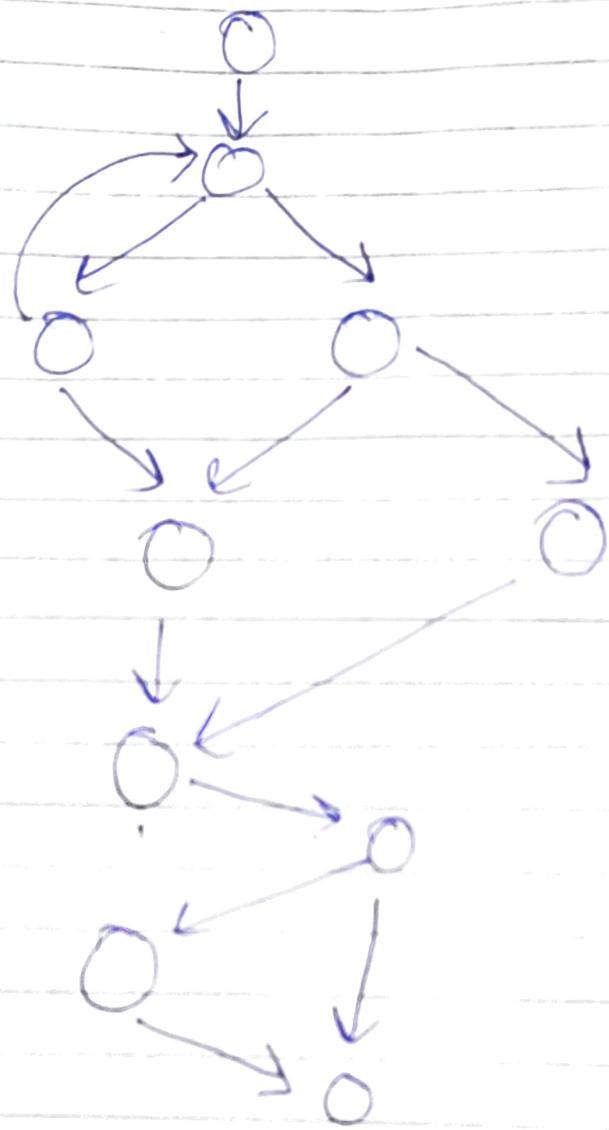
$$(1) V(G) = E - N + 2P$$

$e$  = edges,  $V(G)$  = Cyclomatic Complexity  
 $n$  = nodes,  $P$  = Connected Components

$$(2) V(G) = \pi + 1 \quad \pi = \text{Predicate nodes}$$

$$(3) V(G) = \text{no. of regions} + 1$$

for ex -



Let solve this through FT formula

$$\nabla(G) = e - n + 2P$$

$$\begin{aligned} \nabla(G) &= 13 - 10 + 2 \times 1 \\ &= 5 \end{aligned}$$

## Q6 Explain Manual & Automation Testing

Sol Manual testing is a testing where the tester can test the application without any knowledge of any programming language.

The test engineer tests the application like a user to make it bug-free or stable.

It always search for the fault in the product before the produced released in the market.

The tester execute test cases & generate test report without any help from the automation tools.

## Automation Testing -

It is a type of testing in which we take the help of tools to perform the testing. It is faster than manual testing because it is done with some automation tool. There is no ~~need~~ change of any human error.

It is used to increase the efficiency & effectiveness.

It uses automation testing tools to automate the manual design test cases without any human interfaces.

Anti

Dis  
Managed

It is reliable

not reliable

faster

time consuming

Investment is required  
for tools & automated  
engineers

Investment is required  
for human resources

batch testing is  
possible

not possible

## ~~Q7~~ Software Quality Assurance with ex.

~~SQA~~  
It is simply a way to assure quality in the software. It is a set of activities which ensures processes, procedures & implemented correctly.

- The software Quality assurance is an umbrella activity for checking the quality of the s/w, whether the software we are making is quality or not
- SQA is not checked at any particular phases, it runs along with software development after every development process, software quality assurance is checked

(iii) Generally the quality of the software is verified by the third party organization like ISO (International Standard Organization) to ensure that the software must contain a quality product & it working according to user requirement

(iv) A Software Quality Product has several quality factors.

- Reliability
- functionality
- Error free
- Maintainability
- Usability

Example -

Suppose a SDT is working on a banking application that allows users to manage their accounts, transfer funds, & perform various financial transactions. To ensure the ongoing reliability of the application as new feature are added the QA team employs automated regression testing.

Q8

Statement Coverage, & Conditional Coverage with its formula & Applications.

Ans

Statement Coverage This technique used to design white box test cases.

- If it is also known as line coverage
- It measures the percentage of executable statements in the code that have been executed during testing
- It indicates how much of the source code has been tested by the executed test cases

$$\text{Statement Coverage} = \frac{\text{No. of Executed stat.}}{\text{Total no. of statement}} \times 100\%$$

```
def add_number(a, b):
```

```
    result = a + b
```

```
    print(result)
```

```
    return result
```

100%.

Application -

- It identify by indicating which part of code have been tested which have not
- It is useful for determining the overall quality of test & identify potential gap in testing
- useful for identifying dead code



## Condition Coverage

Dt. \_\_\_\_\_  
Pg. \_\_\_\_\_

- It is used to cover all conditions.
- It is also known as ~~more~~ decision coverage in which each one of the Boolean expression have been evaluated to both True and False

$$\text{Cond'n Coverage} = \frac{\text{No. of executed branches} \times 100\%}{\text{Total no. of Branches}}$$

### Applications →

- It ensures that all possible branches of the code are tested
- It is particularly useful for testing complex control flow structures (such as nested if else) or switch case
- useful for identifying areas where certain code paths are not being exercised.

## CMM Model with applications Pg. No. 1

Sol: It is a process used to develop & refine an organization's software development process.

- The CMM provide the different levels based on the standards. If company acquires so if an IT company is a new company then the CMM will provide "level 1" like this if the company start developing & following the guidelines provided by CMM to increase the maturity level.

The CMM provides total 5-levels -

- i) initial
- ii) Repeatable
- iii) Defined
- iv) Managed
- v) Optimizing

Applications →

- It helps in Predicting Project Outcomes more accurately.
- Organizations regularly assess their processes, learn from mistakes & find way to improve & deliver higher quality software.
- CMM help Software Development team become better at making software. The team can make sure the software works well and is done on time.

## Q10 Dataflow Testing

Dr.

Pg.

- If it is one of the white box testing
- It focuses on two points -
  - in which statement the variable are defined
  - in which statement the variable are used
- It design test cases that cover control flow path around variable definition & their uses in the modules.

Ex →	read a,b,c; if (a>b) n = a + 1 'print n'; else n = b - 1 'print n';	Defn	Define	use
		a	1	2,3
		b	1	2,5
		c	1	NA
		n	3,5	4
		2	NA	6

## Control flow Testing

It is also one of the white box testing.

- It focuses on exercising the program's control structure such as loops, conditionals (if-else) & branch point to ensure that all possible control paths are tested. The goal is to verify that the program behaves as expected under different control flow scenarios.

```

Code def check_num(x):
    if n > 0:
        print("Positive");
    elif n == 0:
        print("zero");
    else:
        print("negative")
    
```

In this we make sure we test every possible outcome of our program.

We need to check 3 scenarios when  $n$  is true,  $n$  is zero &  $n$  is -ve

Control flow is about exploring every route through the program to make sure we test all our program work correctly.

~~Q11~~

Review - A review is when a team of developer examine of piece of software related work & doc such as code, requirement documents. The purpose is to identify errors suggest improvement

Inspection → An inspection is defined as formal, in depth group review designed to identify problems as close to their point of origin as possible. It is initiated by project team

Walkthrough - It is a method of conducting informal group / individual review. Author describes & explain work product in a informal meeting to his supervisor to get feedback. It is cheap to make changes.

## Q2 Check Points with examples.

~~Sol~~ It is a stage during a project where team take a break to assess programs, address issue & make decision what to do next.

1. Planning Check Point - teams plan out what they need to do.  
ex - before developing a new app, team gather to discuss what feature the app should have, target audience.

2. Design Check Point - teams create a design for software how it will look  
ex - before building a website, the team create frame of each page & decide color scheme & font to use

3. Development Check Point - Team write the code & build the software  
ex - while creating a game, developer check that the code is written correctly.

4. Testing checkpoint - After Development team tests the software to identify & fix any bugs. before releasing it to users

Ex- before launching new version of mobile app. It ensures all features work as expected no crashes

5. Deployment checkpoint - once the software has been tested & approved, it is release to users.

Ex - the team deploys the latest version of the s/w to ~~as~~ the app store, making it available for user to download

Q13

Explain equivalence partitions with examples

~~Ans~~ If in a black box testing, It is a software testing technique that divides the input data of a software application into partition of data that are considered equivalent

The idea is to reduce the no of test cases by selecting representative value from each partition

An equivalence class represent a set of valid or invalid states for an input condition.

## Guidelines.

- If range of condn is input then one valid & two invalid equivalence class are different
- If a specific value input one valid & two invalid
- If member of set. input then one valid & one invalid
- If boolean no input " " " " " " "

Ex → There is a college that gives admission to student based on percentage. If will be accepted percentage between 50 to 90%. more & even less than not accepted. If percentage entered by user is less than 50% or more than 90%. Then equivalence Partitioning method will show invalid percentage. If percentage entered is b/w 50 - 90%, then equivalence partitioning method will show valid percentage

Equivalence Partitioning		
Invalid	Valid	Invalid
$x = 50$	$50 - 90$	$y = 90$

↓  
Accept %. Value b/w 50-90