M. Tech Solution (2024-25)

Software Testing & Auditing

MTCS-022

(Section-A)

a) Explain how does test design help in identifying defects.

Ans Creating a structured plan and strategy to thoroughly test s/w, ensuring that all features and functionalities are covered, including edge cases, which helps uncover potential issues before release.

b) What is boundary value analysis (BVA)?

Ans Boundary value analysis is a black box testing techniques which include min, min+1, nominal, max-1, max.

c) What is the main limitation of black box testing?

Ans
- limited test coverage.
- Inability to test complex systems
- Higher cost and time consumptions
- Lack of reusability

d) Explain the primary goal of integration testing?

Ans. The primary goal of integration testing is to verify that different software modules or components, previously tested individually,

interact and function correctly as a cohesive system.

e) Illustrate the essential components of a review plan.

Ans
- Timeline and Resources
- Participants and Roles.

f) What are the different types of review conducted in software testing?

Ans
- Formal Review
- Informal Review

g) Illustrate the role of recorder in a software audit review?

Ans The recorder in a software audit review is responsible to make a written documents of interviews and reviews.

(Section-B)

2.

a) Describe the role of a software tester in a development organization. How does a tester collaborate with developers and other stakeholders to ensure software quality?

Ans In a development organization, a software tester's role is to ensure software quality by identifying and reporting defects, verify functionality, and validating performance against requirements, ultimately contributing to a polished and reliable product.

Collaborate with developers and stakeholders to ensure quality by maintaining open, transparent communication. Hold regular meetings to discuss requirements, progress, and issues. Share detailed test plans and results and provide constructive feedback.

Explain the white box approach to test case design. How does it differ from black box testing?

Techniques used in white box testing

a) Mutation Testing

b) Basic Path Testing

Mutation Testing

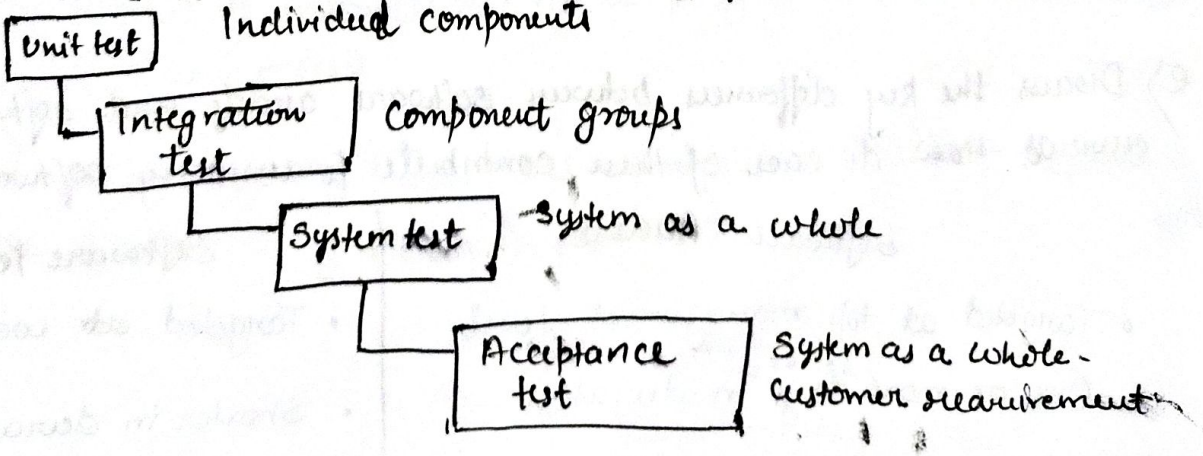
- Mutation Testing is a white box fault-based testing technique that provide strong quality assurance.
- Mutation testing is a method of software testing, which involves modifying source code in small ways

Basic Path testing

- This technique is based on the control structure of a program.
- In this technique control flow graph of a program is prepared and then value of cyclomatic complexity (CC) and no of independent paths are identified.

White box	Black box
• Mostly done by software ~~testers~~ developers	• Mostly done by software ~~developers~~ testers
• It is also called glass box testing	• It is also called behavioural testing.

c) Illustrate the need for different levels of testing in software development How do these levels contribute to s/w quality?



Execution-based software testing, especially for large system, is usually carried out at different levels.

- In most cases there will be 3-4 levels, of testing: unit test, integration test, system test and acceptance test.
- Each of these may consist of one or more sublevels or phases.
- At each level there are specific testing goals.
- A principal goal is to detect functional and structural defect in the unit.

d) Differentiate between testing and ~~Hebb~~ debugging. How do they contribute to software quality?

Testing	Debugging
• It is the process using which we find errors and bugs	• It is the process of finding and fixing the errors and bugs.
• We can identify the failure of any implemented code using this process	• We use this process to provide the code failure with an absolution.
• Errors get displayed in this process	• Errors get deducted and dissolved in this process
• It is a stage of SDLC	• It's not at all an aspect of the SDLC.

Testing identifies defects and verify software against requirement, while debugging locates and fixes those defects in the code.

e) Discuss the key differences between software audits and software peer reviews. How do each of these contribute to ensuring software quality?

Ans

Software Audits	Software Peer review
• Targeted at top management level	• Targeted at coorker level
• One or more days in duration	• Shorter in duration

...us on management system as a visual/observational aspect but also gathers evidence from documentation	• Focus on workplace • visual / observational.

Software audits and previews play a crutial role in ensuring s/w quality by identifying potential issues early, ensuring compliance with standards, and improving overall product.

(Section-C)

3)

a) Define key terms in software testing such as test case, test plan, test scenario, and test script. How do these elements contribute to systematic software testing?

Ans Test Case

A test case is a detailed, documented set of instruction designed to verify that a specific feature or functionality of a s/w application works as expected.

Test Scenario.

Test Scenario is a high-level description of a unit interaction

Test plan

A test plan is a comprehensive document that outlines the overall strategy.

Test Script

A test script is a set of instructions or a program that automates the execution of a test case.

b) What is a defect repository? How does maintaining a defect repository improve the effectiveness of software testing?

Ans A defect repository, also known as bug tracking system or issue tracking system, is a central database used to store manage and track defects. Maintaining a defect repository improves slw testing effectiveness by providing a centeralized, organized system for tracking, managing and analyzing defects, enabling better communication, prioritization and root cause analysis, ultimately leading to faster resolution and improved slw areality.

4)

a) What is eauivalence class partitioning? Explain how this technique is used in black box test case design with suitable ~~design~~ example.

Ans Eauivalence partitioning is a technique of slw testing in which input data is divided into partitions of valid and invalid values, and it is mandatory that all partition must exhibit the same behaviour.

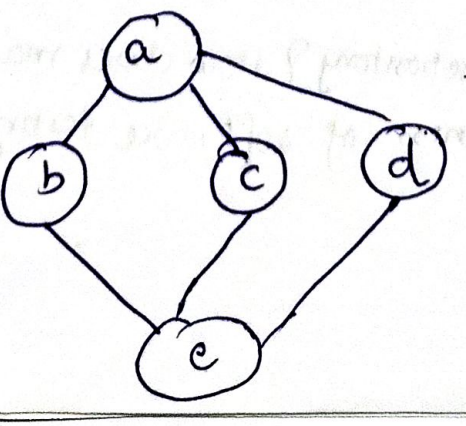
Example:-

1. OTP Number = 6 digits.

INVALID	INVALID	VALID	VALID
1 Test Case	2 Test Case	3 Test Case	
Digit >=7	Digit <=5	Digit =6	Digit =6
9387262	9845	426234	451483

b) What is path coverage in white box testing? Explain its role in ensuring that all code paths are tested adeauately.

Ans. Path coverage ensures that every possible execution path within the code is tested, aiming to identify potential bugs or inefficiencies in complex logic



Cyclomatic complexity = E-N+2

= 6-5+2

= 3

5)

a) Explain regression testing. Also explain how it helps in maintaining s/w stability after modification.

Ans Regression Testing is a s/w testing process where you re-run existing tests after making changes to ensure that new code modifications haven't introduced any unintended bugs or broken existing functionality. Regression testing helps maintain s/w stability after modifications by ensuring that new changes haven't introduced unintended issues.

b) Discuss the concept of system testing, Explain the different types of system testing and their importance in s/w validation.

Ans. It is the third level of testing

- Here, system is tested as a whole
- It has two types. They are :-
  - → Functional Testing
  - → Non-Functional Testing

Functional Testing

→ Here, tests are run on the structure of the project

→ It is done by developers & testers to know or find out the error

Non-functional Testing

→ Here, it is done to check the tolerance of the project over steps, security etc

→ It is done to know whether the project is durable & serve or handle the load.

6)

a) Explain the role of SCM in s/w testing. How does effective SCM contribute to test planning and execution?

Ans = SCM plays a crucial role in s/w testing by ensuring that the correct version of code, documentation and other artifacts are used for testing, facilitating effective tracking of changes to maintain s/w quality.

1) Version Control

SCM helps establish and manage baselines for testing purposes

2) Change Management.

SCM implements procedures to control changes to the s/w, ensuring that changes are properly planned, tested and integrated.

3) Test Environment Management.

SCM helps ensure that test environments are consistent with the s/w versions being tested, which is essential for repeatable testing.

b) Explain the significance of test planning in s/w development. What key factors must be considered while preparing a test plan?

Ans. Test planning is critical in s/w development as it provides a structured approach, ensuring comprehensive testing, ultimately leading to higher quality s/w

- Ensures Comprehensive Testing
- Provides Clarity and Direction.
- Facilitates Risk Management
- Improves Resource Allocation
- Support Quality Assurance

7)

a) What are the key objectives of a software audit? How does conducting an audit improve the s/w development process and product quality?

mpliance

Audits verify that s/w development practices and the final product adhere to relevant industry standards, regulations and legal requirements.

- Security

Audit verify that the s/w development must be secure from the hackers

- Quality

Audits assess the quality of code, identifying areas for improvement in terms of readability, maintainability, and performance

Process Improvement

Audits identify inefficiencies in the s/w development process and suggest improvements to streamline workflows.

Documentation

Audits verify that the s/w is well-documented, making it easier to maintain and update.

Code Reduction

Audits help identify issues early in the development lifecycle, reducing the cost of fixing them later.

b) What are software management reviews? Explain their objectives and the role they play in the overall s/w development process.

Ans A Software Management Review (SMR) is a management-level assessment of a s/w project's status and resource allocation, distinct from technical reviews, or audits, focusing on project health and alignment with goals.

- Early Issue Detection
- Progress Monitoring
- Resource Allocation
- Risk Mitigation
- Stakeholder Alignment
- Decision-Making
- Improvement.