UT Solution

Software Testing & Auditing

MTCS-022

(2023-24)

Section-A

Q1.(a) Explain the need of software testing.

Ans1.(a) Software testing identifies bugs & issues in the development process so they're fixed prior to product launch.

Q1.(b) Discuss the origins of defects in software development

Ans1.(b). Defects in software are inevitable. In fact, more defects are caused by poor requirements, poor architecture or poor design than poor code.

Q1.(c) Define black-box testing.

Ans1.(c) Black box testing is a technique in which the internal workings of the software are not known to the tester.

Q1.(d) Discuss the different types of system tests in software development.

Ans1.(d) Different types of system tests in software development are performance testing, load testing, stress testing & scalability testing.

Q1.(e) Define the concept of "levels of testing" in software development.

Ans1.(e) In software testing, there are four levels of testing: Unit testing, Integration testing, System testing & Acceptance testing.

Q1.(f) Explain the purpose of test plan attachments.

Ans1.(f) The details needed for organizing & executing the tests, for eg- what are the required inputs outputs & procedural steps for each test, where will the tests be stored for each item.

Q1.(g) Name two participants involved in a software audit review.

Ans1.(g) Two participants involved in a software audit review are auditor & recorder.

Section - B

Q2.(a) Define software testing principles & explain how they guide testing activities in a software development project.

Ans2.(a) Different software testing principles are:-

→ Testing shows the presence, not the absence of error states.

→ Complete testing is impossible.

→ Early testing saves time & money.

→ Tests wear out.

→ Testing is context-dependent.

→ Fallacy: "No errors" mean a usable system.

Software testing principles are the core theoretical concepts that underpin the entire process & practice of software testing. They ensure that testing is conducted

ficiently, effectively & is alligned with achieving the intended outcomes.

Q2 (b) Explain the concept of Equivalence class partioning & boundary value analysis.

Ans2 (b) Equivalence class partioning is a black box testing where a range of input values are divided into equivalence data classes. In this, the tester tests a random input value from the defined interval of equivalence data classes & if the output for that input value is valid, then the whole class interval is considered valid & vice-versa.

Ex→ An application allow the user to enter the password of length 8-12 number

Invalid Equivalence Class	Valid Equivalence Class	Invalid Eq Class
<8	8-12	>12

BVA is used to check the behaviour of application using test data that exist at boundary values. It is mostly used in design technique as it is believed that software is most likely to fail at upper & lower limits of input data values.

Q2.(c) Why is there a need for different levels of testing in software development? How do these levels complement each other to ensure the overall quality of the software product?

Ans2.(c) A level of software testing is a process where every unit or component of a software or system is tested. The main reason for implementing the levels of testing is to make the software testing process efficient & easy to find all possible test cases at a specific level.

these levels complement each other & they ensure the overall quality of the software product. The main idea behind this concept is that each level of testing targets specific aspects of the software functionality, which allows for better quality assurance & fewer potential defects. The most common types of testing levels include - unit, integration, system & acceptance testing.

(Q2-cd) What are the fundamental concepts of testing, & why is testing essential in the software development process?

Ans 2-cd) Testing is a process of executing a program or part of a program with the intention of finding errors. The different phases of a test life cycle

are Test Planning & Control, Test Analysis & Design, Test Implementation & Execution, Evaluation Exit Criteria & Reporting & Test Closure.

Testing is essential in software development process as it ensures conformance to business needs & technical specifications. Testing also identifies defects & flaws in the software early in the development lifecycle when they are less expensive to fix. The later a bug is found, the costlier it becomes to resolve.

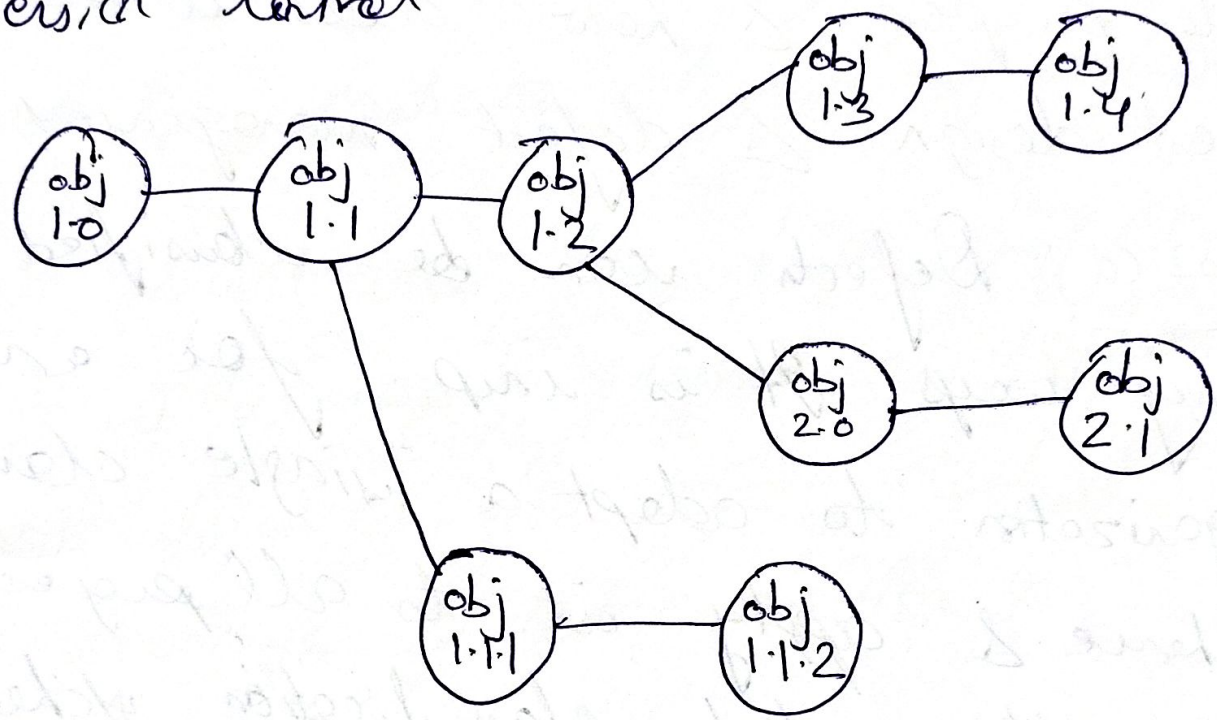
Q2.(e) How does software configuration Management (SCM) contribute to the testing process?

Ans2(e). SCM is an arrangement of exercises that controls change by recognizing the items for change, setting up connection b/w those things,

ing instruments for overseeing diverse aspects, controlling the changes being executed. Process involved in SCM -

→ Identification & Establishment

→ Version Control



→ Change Control

→ Configuration auditing

→ Reporting

SCM involves a set of process & tools that help to manage the different components of a software system, including source code, documentation & other assets.

Section - C

Q3.(a) Explain the concept of a defect repository in software testing. What is its purpose, & how does it aid in test design & defect management?

Ans3.(a) Defects can be classified in many ways. It is imp. for an organization to adopt a single classification scheme & apply it to all projects. No matter which classification scheme is selected, some defects will fit into more than one class or category. Because of this problem, developers, testers & SQA staff should try to be as consistent as possible when recording defect data.

the defect types & frequency of occurence should be used in test planning & test design. Execution based testing strategies should be selected that have the strongest possibility of detecting particular types of defects.

Q3.c) What are the basic definitions of terms commonly used in software testing such as test case, test plan test coverage & test execution ?

Ans 3.c) Test case is a set of preconditions, inputs, actions, expected results & post conditions, developed based on test conditions.

Test plan is a document that describes the scope, approach, resources & schedule of intended test activities.

Test coverage is defined as a metric in software testing that measures the amount of testing performed by a set of test. In simple terms, it is a technique to ensure that your tests are letting you code or how much of your code you exercised by running the test.

Test execution is a process of running test cases based on test scenarios created for software applications to ensure that it meets all the pre-defined functional & non-functional requirements or specifications.

Ans 4. (a) Explain the concept of coverage & control flow graphs in white-box testing. How do these concepts aid in measuring the completeness of test coverage?

Ans 4. (a) A control flow graph is the graphical representation of control flow or computation during the execution of programs or applications. CFG are mostly used in static analysis as well as compiler applications, as they can accurately represent the flow inside a program unit.

→ The CFG is process oriented.

→ The CFG shows all the paths that can be traversed during a program execution.

→ A CFG is a directed graph.
→ Edge in CFG portray control flow paths and the nodes in CFG portray basic blocks.

Q4.(b) Explain the concept of random testing. What are its advantages & limitations compared to other test case design strategies.

Ans 4.(b) Random testing is software testing in which the system is tested with the help of generating random & independent inputs & test cases. It is also named as monkey testing. It is implemented when the bug in an application is not recognized.

Advantages :
→ It is very cheap so that anyone can use this software.

(c) It doesn't need any special intelligence to access the program during the tests.

Disadvantages -)

→ This is not practical. Some tests will be of no use for a longer time.

→ Most of the time is consumed by analyzing all the tests

→ New tests cannot be found if their data is not available during testing.

Q5(a) Discuss integration testing. What is its purpose & how does it differ from unit testing?

(Ans 5. (a)) Integration testing is the process of testing the interface b/w two software units or modules. It focuses on determining the

correctness of the interface.
The purpose of integration testing is to expose faults in the interaction between integrated units. Once all the modules have been unit-tested, integration testing is performed.

Unit Testing	Integration Testing
→ It focuses on individual units or components	→ It focuses on interactions b/w units.
→ It is performed first.	→ It is performed after unit tests.
→ It is executed usually by the developers	→ It is executed usually by a separate tester

Q5 (b) Define regression testing & discuss its importance in software development. How does regression testing help maintain the integrity of the functionality ?

Ans 5.(b) Regression testing is the process of testing the modified parts of the code & the parts that might get affected due to the modifications to ensure that no new errors have been introduced in the software after the modifications have been made.

Regression testing is a significant step in the software development process overall. It helps to ensure that changes to the system do not cause any unexpected errors, verify that functionality has not been lost & uncover previously unknown defects. It ensures integrity & checks that updates don't harm the products functionality. It also ensures that new changes don't introduce unexpected issues.

Q6 (a) What are the primary goals of testing in software development?

Ans 6 (a) The primary goals of software testing are -

→ Always identifying the bugs as early as possible.

→ Preventing the bugs in a project & product.

→ Check whether the customer requirements criteria is met or not.

→ Finally main goal is to measure the quality of the product & project.

The main goal is to show that application is working as per the requirements defined by the client.

6.cb) Explain the significance of measurements & milestones in controlling & monitoring the testing process.

Ans 6.cb) All process should have measurements (metrics) associated with them. The measurements helps to answer questions about status & quality of the process as well as the products that result from its implementation. Measurement can help to track test progress evaluate the quality of the software product, manage risks, classify & prevent defects & evaluate test effectiveness.

Milestones are tangible events that are expected to occur at a certain time in the project's lifetime. Managers

use them to determine project status. It can be used to monitor the progress of the testing efforts associated with a software project.

Q7. a) Discuss the importance of ongoing learning & knowledge sharing in enhancing the effectiveness of audit activities.

Ans7. (a) Successful knowledge management projects are rooted in a compelling business need & a clear objective to enhance operational effectiveness within organizational unit. Within the assurance & an audit industry, the knowledge staff assumes a pivotal role in capture, sharing, distribution & application critical expertise

evaluate the quality, efficiency & overall effectiveness of audit process. In auditing activities, the economic engine lies in leveraging the professional knowledge held by employees. The final product delivered to clients is not merely an audit report but a service that satisfies clients & stakeholders.

(Q7-b) What are the key differences in objectives between software audits & peer reviews?

Ans7-b) A software audit is a type of s/w review in which one or more auditors who are not members of the s/w development organization conduct "An independent examination of a software product, s/w process,

Or a set of s/w process to assess compliance with specifications, standards, contracted agreements or other criteria".

It is different from software peer review in that they are conducted by personnel external to s/w independent of the software development organization & are concerned with compliance of products or processes rather than with their technical content, technical quality or managerial implications.