IT2032 SOFTWARE TESTING

UNIT – 1

1. Define process.

Process, in the software engineering domain, is the set of methods, practices, standards, documents, activities, policies, and procedures that software engineers use to develop and maintain a software system and its associated artifacts, such as project and test plans, design documents, code and manuals.

2. Define Validation.

Validation is the process of evaluating a software system or component during, or at the end of, the development cycle in order to determine whether it satisfies specified requirements.

3. Define Verification.

Verification is the process of evaluating a software system or component to determine whether the product of a given development phase satisfy the conditions imposed at the start of that phase.

4. Define Testing.

Testing is generally described as a group of procedures carried out to evaluate some aspects of a piece of software.

(OR)

Testing can be described as a process used for revealing defects in software, and for establishing that the software has attained a specified degree of quality with respect to selected attributes.

5. Define Debugging.

Debugging, or fault localization is the process of

- ✓ Locating the fault or defect.
- ✓ Repairing the codes.
- ✓ Retesting the code.

6. List out the levels of the testing maturity model..

- ✓ Level 1: Initial.
- ✓ Level 2: Phase Definition.
- ✓ Level 3: Integration.
- ✓ Level 4: Management and Measurement.
- ✓ Level 5: Optimization/Defect prevention and quality control.

7. Define Errors.

An error is a mistake, misconception, or misunderstanding on the part of a software developer.

8. Define Faults.

A fault (defect) is introduced into the software as the result of an error. It is an anomaly in the software that may cause it to behave incorrectly, and not according to its specification.

9. Define Failures.

A failure is the inability of a software system or component to perform its required functions within specified performance requirements.

10. Define Test case.

A test case in the practical sense is a test- related item which contain the following information:

- ✓ A set of test inputs.
- ✓ Execution conditions.
- ✓ Expected outputs.

11. Define Test.

A test is a group of related test cases, or a group of related test cases and test procedures.

12. Define Test Oracle.

A test oracle is a document, or piece of software that allows tester to determine whether a test has been passed or failed.

13. Define Test Bed.

A test bed is an environment that contains all the hardware and software needed to test a software component or a software system.

14. Define Quality.

Two concise definitions for quality.

- ✓ Quality relates to the degree to which a system, system component, or process meets specified requirements.
- ✓ Quality relates to the degree to which a system, system component, or process meets customer or user needs, or expectations.

15. Define Metric.

A metric is a quantitative measure of the degree to which a system, system component, or process possesses a given attribute.

16. Define Quality Metric.

A quality metric is a quantitative measure of the degree to which an item possesses a quality attribute.

17. Define SQA.

The software quality assurance (SQA) group is a team of people with the necessary training and skills to ensure that all necessary actions are taken during the development process so that the resulting software conforms to established technical requirements.

18. Define Review.

A review is a group meeting whose purpose is to evaluate artifact or a set of software artifacts.

19. Define Precondition.

A precondition is a condition that must be true in order for a software component to operate properly.

Eg; number_of_coins > =0

20. Define Post condition.

A post condition is a condition that must be true when a software component completes its operation properly.

Eg; number_of_dollars, number_of_cents > = 0

UNIT – 2

1. List the two basic Testing strategies.

- ✓ Black box testing.
- ✓ White box testing.

2. What are the knowledge sources for Black box testing?

- ✓ Requirements
- ✓ Document specification
- ✓ Domain knowledge
- ✓ Defect analysis data
- ✓

3. What are the knowledge sources for White box testing?

- ✓ High level design
- ✓ Detailed design
- ✓ Control flow graphs
- ✓ Cyclomatic complexity

4. List the methods of Black box testing.

- ✓ Equivalence class partitioning
- ✓ Boundary value analysis
- ✓ State transition testing
- ✓ Cause and effect graphing
- ✓ Error guessing

5. List the methods of White box testing.

- ✓ Statement testing
- ✓ Branch testing

- ✓ Path testing
- ✓ Data flow testing
- ✓ Mutation testing
- ✓ Loop testing

6. Define Random testing.

Each software system or module has an input domain from which test input data is selected. If a tester randomly selects input from the domain, this is called Random testing.

7. Define Equivalence class partitioning.

If a tester is viewing the software-under-test as a black box with well defined inputs and outputs, a good approach to selecting test inputs is to use a method called Equivalence class partitioning.

8. List the advantages of Equivalence class partitioning.

- ✓ It eliminates the need for exhaustive testing, which is not feasible.
- ✓ It guides a tester in selecting a subset of test inputs with a high probability of detecting a defect.
- ✓ It allows a tester to cover a larger domain of inputs/outputs with a smaller subset selected from an Equivalence class.

9. Define Cause effect graphing.

It is a technique that can be used to combine conditions and derive an effective set of test cases that may disclose inconsistencies in a specification.

10. Define State.

A state is an internal configuration of a system or component. It is defined in terms of values assumed at a particular time for the variables that characterize the system or component.

11. Define Finite-state machine.

It is an abstract machine that can be represented by a state graph having a finite number of states and a finite number of transitions between states.

12. Define Usage profiles.

Usage profiles are characterizations of the population of intended uses of the software in its intended environment.

13. What is Certification

Certification refers to third-party assurance that a product, process, or service meets a specific set of requirements.

14. What is Test data set?

A test data set is statement, or branch, adequate if a test set T for program P causes all the statements, or branches, to be executed respectively.

15. Define Path.

A path is a sequence of control flow nodes usually beginning from the entry node of a graph through to the exit node.

16. Define Variable.

Variable is defined in a statement when its value is assigned or changed.

(OR)

Variable is defined in a statement when its value is utilized in a statement. The value of the variable is not changed.

17. List the two major assumptions in Mutation testing.

- ✓ The component programmer hypothesis
- ✓ The coupling effect

18. Define Test set.

A test set T is said to be mutation adequate for program P provided that for every in equivalent mutant Pi of P there is an element t in T such that Pi[t]is not equal to P[t].

19. Define Error guessing.

Error guessing approach is based on the testers/developers past experience with code similar to code-under-test, and their intuition as to where defects may lurk in the code.

20. What is the goal of smart tester?

The goal of the smart tester is to understand the functionality, input/output domain, and the environment of use for the code being tested.

UNIT – 3

1. List the different levels of testing.

- ✓ Unit test
- ✓ Integration test
- ✓ System test
- ✓ Acceptance test.

2. Define Unit Testing

A unit is the smallest possible testable software component that can be characterized in several ways.

3. List the components suitable for unit test.

- ✓ Procedures and functions
- ✓ Classes/objects and methods
- ✓ Procedure-sized reusable components.

4. List the phases in the unit test planning.

- ✓ Phase 1: Describe unit test approach and risks.
- ✓ Phase 2: Identify unit features to be tested.
- ✓ Phase 3: Add levels of detailed to the plan.

5. List the issues in the unit test.

- ✓ Issue 1: Adequately testing classes.
- ✓ Issue 2: Observation of objects states and state changes.
- ✓ Issue 3: The retesting of classes-I
- ✓ Issue 4: The retesting of classes-II

6. What is Test harness?

The auxiliary code developed to support to testing of units and components is called a test harness. The harness consists of drivers that call the target code and stubs that represent modules it calls.

7. List the major goals of Integration test.

- ✓ To detect defects that occurs on the interfaces of units.
- ✓ To assemble the individual units into working subsystems and the finally a complete system that is ready for system test

8. What is the advantage of Bottom up integration?

Bottom-up integration has the advantage that the lower-level modules are usually well tested early in the integration process. This is important if these modules are candidates for reuse.

9. What is a cluster?

A cluster consists of classes that are related, for example, they may work together to support a required functionality for the complete system.

10. List the several types of system tests.

- ✓ Functional testing
- ✓ Performance testing
- ✓ Stress testing
- ✓ Configuration testing
- ✓ Security testing
- ✓ Recovery testing

11. Define Load.

A load is a series of inputs that simulates a group of transactions.

12. List the two major requirements of Performance testing.

- ✓ Functional requirements
- ✓ Quality requirements.

13. What is meant by Stress testing?

When a system is tested with a load that causes it to allocate its resources in maximum amounts, this is called stress testing.

14. Give the examples of security testing.

- ✓ Password checking
- ✓ Legal and illegal entry with password
- ✓ Password Expiration
- ✓ Encryption
- ✓ Browsing
- ✓ Trap doors
- ✓ Viruses.

15. Define Recovery testing.

Recovery testing subjects a system to losses of resources in order to determine if it can recover properly from these losses.

16. List the areas covered during recovery testing.

- ✓ Restart
- ✓ Switchover.

17. Define Use case.

A use case is a pattern, scenario, or exemplar of usage. It describes a typical interaction between the software system under development and a user.

18. Define Regression testing.

Regression testing is not a level of testing, but it is the retesting of the software that occurs when the changes are made to ensure that the new version of the software has retained the capabilities of the old version and that has no defect have been introduced due to the changes.

19. List the objectives of configuration testing.

- ✓ Show that all the configuration changing commands and menus work properly
- ✓ Show that all interchangeable devices are really interchangeable, and that they each enter the proper states for the specified conditions
- ✓ Show that the system's performance level is maintained when devices are interchanged, or when they fail.

20. List the effect of security breaches.

- ✓ Loss of information
- ✓ Corruption of information
- ✓ Misinformation
- ✓ Privacy violations
- ✓ Denial of service.

UNIT – 4

1. Define Goal in testing

A Goal can be described as

A statement of intent

A statement of a accomplishment that an individual

2. What are the three types of goals in testing

- ✓ Business Goal
- ✓ Technical Goal
- ✓ Political Goal

3. Define the term policy

A policy can be defined as a high-level statement of principle or course of action that is used to govern a set of activities in an organization.

4. Define Test Plan

A Plan is a document that provides a frame work or approach for achieving a set of goals.

5. List the various Test Plan components

- ✓ Test Plan identifier
- ✓ Introduction
- ✓ Items to be tested
- ✓ Features to be tested
- ✓ Pass/Fail criteria
- ✓ Suspension & Resumption criteria
- ✓ Testing tasks
- ✓ Test environment
- ✓ Risks & Contingencies
- ✓ Testing costs
- ✓ Approvals

6. Define Features

Features may be described as distinguishing characteristics of a software component or system.

7. Define the term Pass / Fail Criteria

Given a test item and a test case, the tester must have a set of criteria to decide on whether the test has been passed or failed upon execution.

8. Define Suspension & Resumption criteria.

The criteria to suspend and resume testing are described in the simplest of cases testing is suspended at the end of a working day and resumed the following morning.

9. Define Work Breakdown Structure (WBS)

A Work Break Down structure is a hierarchical or tree like representation of all the tasks that are required to complete a project.

10. Define Risks & Contingencies.

Every testing effort has risks associated with it. Testing software with a high degree of critically, complexity, or a tight delivery deadline all impose risks that may have negative impacts on project goals.

11. Define Cost Driver

A Cost Driver can be described as a process or product factor that has an impact on overall project costs.

12. Explain the simple COCOMO equation

$E = a (\text{size in KLOC})^b$

13. What are the various components of the test plan

- ✓ Test Design Specification
- ✓ Test Case Specification
- ✓ Test Procedures specifications

14. Define Test Procedure

A Procedure in general as a sequence of steps required to carry out a specific task

15. Define Test Summary Report

This report is prepared when testing is complete. It is summary of the results of the testing efforts. It also becomes a part of the projects historical database and provides a basis for lessons learned as applied to future projects.

16. List the skills needed by a Test specialist

- ✓ Organizational and planning skills
- ✓ The ability to keep track of and pay attention to details
- ✓ The determination to discover and solve problems
- ✓ The ability to mentor and train others
- ✓ The ability to work with users and clients
- ✓ The ability to think creatively

17. What are the steps in forming the test group.

- ✓ Upper management support for test function
- ✓ Establish test group organization, career paths
- ✓ Define education and skill levels
- ✓ Develop job description
- ✓ Interview candidates
- ✓ Select Test group members

18. Explain the Test team hierarchy

- ✓ The Test Manager
- ✓ The Test Lead
- ✓ The Test Engineer
- ✓ The Junior Test Engineer

19. What is the use of V-model in testing

The V-model is model that illustrates how testing activities can be integrated in to each phase of the standard software life cycle.

20. What are the various approaches to test cost estimation

- ✓ COCOMO Model
- ✓ Use of test cost drivers
- ✓ Test Tasks
- ✓ Testers / Developers ratio
- ✓ Expert judgment

UNIT – 5

1. Define the term Project monitoring.

Project Monitoring refers to the activities and tasks managers engage in to periodically check the status of each project. Reports are prepared that compare the actual work done to the work that was planned.

2. Define the term Project controlling.

Project Controlling consists of developing and applying a set of corrective actions to get a project on track when monitoring shows a deviation from what was planned.

3. Define Milestones

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.

4. List some examples of testing Milestones

- ✓ Completion of the Master test plan
- ✓ Completion of branch coverage for all units
- ✓ Execution of all planned system test
- ✓ Completion of the test summary report.

5. List various Measurements for monitoring testing status.

- ✓ **Coverage Measures**
- ✓ Test Case Development
- ✓ Test Execution
- ✓ Test Harness Development

6. List the types of testing measurements

- ✓ Coverage
- ✓ Test Case Development
- ✓ Test Execution
- ✓ Test Harness

7. What are the various Severity level hierarchy

- ✓ Catastrophic
- ✓ Critical
- ✓ Marginal
- ✓ Minor or Annoying

8. What are the four major activities associated with Configuration management.

- ✓ Identification of the Configuration items
- ✓ Change Control
- ✓ Configuration status reporting
- ✓ Configuration audits

9. Define Change Control Board (CCB).

There are 2 aspects of change control – one is tool based, the other term based. The team involved is called CCB.

10. Define the term Review.

A review is a group meeting whose purpose is to evaluate a software artifact or a set of software artifact.

11. Explain the benefits of review program

- ✓ Higher – quality software
- ✓ Increased productivity
- ✓ Closer adherence to project schedule
- ✓ Increased awareness of quality issues

12. List the types of reviews.

There are two major types of technical reviews

- ✓ Inspections
- ✓ Walkthrough

13. What are the various steps in the inspection process

- ✓ Entry Criteria
- ✓ Initiation
- ✓ Preparation
- ✓ Inspection Meeting
- ✓ Reporting results
- ✓ Rework & follow up

14. Define Walkthrough

Walkthrough are a type of technical review where the producer of the reviewed material serves as the review leader and actually guides the progression of the review. Walkthrough have traditionally been applied to design and code.

15. What are the advantages of review approach.

There are two pass approach for detect detection.

- ✓ Pass 1 has individuals first reading reviewed item
- ✓ Pass 2 has the item read by the group as a whole.

16. What are the various components of review plans.

- ✓ Review Goals
- ✓ Preconditions and items to be reviewed
- ✓ Roles, Participants, Team Size and time requirements
- ✓ Review Procedures
- ✓ Review Training
- ✓ Review Checklist

17. What are the various roles in review program

- ✓ Review Leader
- ✓ Review Recorder
- ✓ Reader
- ✓ Reviewer

18. List the various review team membership constituency

Review Team Members

- ✓ SQA Staff
- ✓ Testers
- ✓ Developers
- ✓ Users / Clients
- ✓ Specialists

19. What are the various different types of software artifacts.

- ✓ Requirement Reviews
- ✓ Design Reviews
- ✓ Code Reviews
- ✓ Test Plan reviews

20. Define Defect Removal Leverage (DRL).

This is a ratio of the defect detection rates from two review or test phases and can be expressed as

$$DRL = \frac{\text{Defects / hour (review or test phase X)}}{\text{Defects / hour (review or test phase Y)}}$$

16-Marks.

UNIT-I

1. Explain in detail about the elements of engineering disciplines.
2. Discuss about the role of process in software quality.
3. Draw the 5-level structure of the testing maturity model and discuss about it.
4. Explain in detail about the software testing principles.
5. Give an example for defect classes and discuss them in detail.

UNIT – 2

Explain in detail about the Equivalence class partitioning.

1. Discuss the various approaches in Black Box test design.
2. Describe the difference between the white box and black box testing strategies.
3. What is a control flow graph? How is it used in white box test design?
4. Explain the differences between random testing and testing using error guessing.

UNIT – 3

1. How would you define a software unit? In terms of your definition, what constitutes a unit for procedural code; for object-oriented code?
2. Discuss the issues that arise in class testing.
3. Why is it so important to design a test harness for reusability?
4. What are the key differences in integrating procedural-oriented systems as compared to object-oriented systems?
5. Describe the activities/Tasks and responsibilities for developer/testers in support of multilevel testing.

UNIT – 4

1. Why is testing planning so important for developing a repeatable and managed testing process?
2. Why is it so important to integrate testing activities into the software life cycle?
3. What role do managers play in support of a test group?
4. Discuss in detail about the test specialist skills.
5. Discuss in detail about the test plan components.

UNIT – 5

1. Discuss in detail about the controlling and monitoring: three critical views.
2. Explain in detail about the role of reviews in testing software deliverables.
3. Discuss in detail about the components of review plans.
4. Explain in detail about the software configuration management.
5. Explain about the various types of reviews.