

---

# IT8076-SOFTWARE TESTING

---

## UNIT – 1

### 1. Define Software Engineering.

Software Engineering is a discipline that produces error free software with in a time and budget.

### 2. 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.

### 3. 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.

### 4. 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.

### 5. 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.

### 6. Define Debugging.

Debugging, or fault localization is the process of

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

### 7. 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.

### 8. List the elements of the engineering disciplines.

- Basic principles
- Processes
- Standards
- Measurements
- Tools
- Methods
- Best practices
- Code of ethics
- Body of knowledge

## IT8076-SOFTWARE TESTING

---

### 9. Differentiate between testing and debugging. (U.Q Nov/Dec 2008)

Testing	Debugging
1. Testing as a dual purpose process <ul style="list-style-type: none"> <li>• Reveal defects</li> <li>• And to evaluate</li> <li>• quality attributes</li> </ul>	1. Debugging or fault localization is the process of <ul style="list-style-type: none"> <li>• Locating the fault or defect</li> <li>• Repairing the code, and</li> <li>• Retesting the code</li> </ul>

### 10. Define process in the context of software quality.

Process, in the software engineering domain, is a set of methods, practices, Standards, documents, activities, polices, 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.

### 11. List the members of the critical groups in a testing process

- Manager
- Developer/Tester
- User/Client

### 12. Define Errors.

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

### 13. 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.

### 14. Define Failures.

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

### 15. 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.

### 16. Define Test.

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

### 17. 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.

### 18. 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.

### 19. Write short notes on Test, Test Set, and Test Suite.

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

A group of related test is sometimes referred to as a test set.

A group of related tests that are associated with a database, and are usually run together, is sometimes referred to as a Test Suite.

---

## IT8076-SOFTWARE TESTING

---

### 20. 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.

### 21. Define Metric.

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

### 22. Define Quality Metric.

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

### 24. Define SQA group.

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.

### 25. Explain the work of SQA group.

Testers to develop quality related policies and quality assurance plans for each project. The group is also involved in measurement collection and analysis, record keeping, and Reporting. The SQA team members participate in reviews and audits, record and track Problems, and verify that corrections have been made.

### 26. Define Review.

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

### 27. List the sources of Defects or Origins of defects. Or list the classification of defect

- Education
- Communication
- Oversight
- Transcription
- Process

### PART-B

1. Explain in detail about the elements of engineering disciplines.

- Principles
- Process
- Standards
- Measurements
- Tools
- Methods
- Best practices
- Code of ethics
- Knowledge

2. Discuss about the role of process in software quality.

- Methods and technics
- Practices policies
- Activities

## IT8076-SOFTWARE TESTING

---

- Standards and documents
  - Plans
  - Procedure
3. Draw the 5-level structure of the testing maturity model and discuss about it.
- Initial
  - Phase definition
  - Integration
  - Management and measurement
  - Optimization/ defect prevention and quality control
4. Explain in detail about the software testing principles.
- a. testing is the process of exercising software component.
  - b. Good test case should have high probability of revealing defect
  - c. Inspect test result meticulously
  - d. Test case should have expected result
  - e. Test case should be developed for both valid and in valid input condition.
  - f. Probability of having additional defect is proportional to already founded defects.
  - g. Testing should be carried by team independent of development.
  - h. Test must be repeatable and reusable
  - i. Testing should be planned
  - j. Testing should be integrated into software lifecycle
  - k. Testing is creative and challenging task.
5. Explain in detail the tester's role in a software development organization
- Plan
  - Execute
  - Record and
  - Analyse test.
6. Explain in detail defect classes, the defect repository and test design
- Requirement and specification defect
  - Design defect
  - Coding defect
  - Testing defect

## IT8076-SOFTWARE TESTING

### UNIT – 2

#### 1. Define Smart Tester.

Software must be tested before it is delivered to users. It is responsibility of the testers to Design tests that (i) reveal defects (ii) can be used to evaluate software performance, usability and reliability. To achieve these goals, tester must select a finite no. of test cases (i/p, o/p, & conditions).

#### 2. List the two basic Testing strategies.

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

#### 3. Compare black box and white box testing.

Black box testing	White box Testing
Black box testing , the tester is no Knowledge of its inner structure(i.e. how it woks)The tester only has knowledge of what it does(Focus only input & output)	The White box approach focuses on the inner structure of the software to be tested.
Black box approach is usually applied large size piece of software	White box approach is usually applied small size piece of software.
Black box testing sometimes called functional or specification testing.	White box sometimes called clear or glass box testing

#### 4. What are the knowledge sources for Black box testing?

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

#### 5. What are the knowledge sources for White box testing?

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

#### 6. List the methods of Black box testing.

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

#### 7. List the methods of White box testing.

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

#### 8. 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.

#### 9. 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.

## IT8076-SOFTWARE TESTING

---

### 10. 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.

### 11. 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.

### 12. List the Knowledge Sources & Methods of black box and white box testing.

Test Strategy	Knowledge Sources	Methods
Black box	<ol style="list-style-type: none"> <li>1. Requirements document</li> <li>2. Specifications</li> <li>3. Domain Knowledge</li> <li>4. Defect analysis data</li> </ol>	<ol style="list-style-type: none"> <li>1. Equivalence class partitioning (ECP)</li> <li>2. Boundary value analysis (BVA)</li> <li>3. State Transition testing.(STT)</li> <li>4. Cause and Effect Graphing.</li> <li>5. Error guessing</li> </ol>
White box	<ol style="list-style-type: none"> <li>1. High level design</li> <li>2. Detailed design</li> <li>3. Control flow graphs</li> <li>4. Cyclomatic complexity</li> </ol>	<ol style="list-style-type: none"> <li>1. Statement testing</li> <li>2. Branch testing</li> <li>3. Path testing</li> <li>4. Data flow testing</li> <li>5. Mutation testing</li> <li>6. Loop testing</li> </ol>

### 13. 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.

### 14. 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.

### 15. Define Usage profiles.

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

### 16. What is Certification?

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

### 17. 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.

### 18. 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.

### 19. 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.

### 20. List the two major assumptions in Mutation testing.

- ✓ The component programmer hypothesis
- ✓ The coupling effect

## IT8076-SOFTWARE TESTING

---

### 21. Define Test set.

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

### 22. 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.

### 23. 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.

### 24. Define COTS Components.

The reusable component may come from a code reuse library within their org or, as is most likely, from an outside vendor who specializes in the development of specific types of software components. Components produced by vendor org are known as commercial off-the-shelf, or COTS, components.

### 25. Define usage profiles and Certification.

Usage profiles are characterizations of the population of intended uses of the software in its intended environment. Certification refers to third party assurance that a product, process, or service meets a specific set of requirements.

### 26. What are the factors affecting less than 100% degree of coverage?

- The nature of the unit
  - Some statements/branches may not be reachable.
  - The unit may be simple, and not mission, or safety, critical, and so complete
  - coverage is thought to be unnecessary.
- The lack of resources
  - The time set aside for testing is not adequate to achieve complete coverage for all of the units.
  - There is a lack of tools to support complete coverage
- Other project related issues such as timing, scheduling. And marketing constraints.

## PART-B

1. Explain in detail about the Equivalence class partitioning.

- Partition input domain of software under test.
- List of conditions:
- Range of values
- Number of values
- Set of valid input values
- Must be condition

2. Discuss the various approaches in Black Box test design.

1. Equivalence class partitioning (ECP)
2. Boundary value analysis (BVA)
3. State Transition testing.(STT)
4. Cause and Effect Graphing.
5. Error guessing

## IT8076-SOFTWARE TESTING

3. Describe the difference between the white box and black box testing strategies.

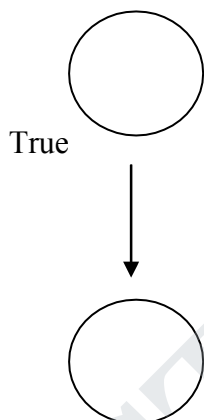
Black box testing	White box Testing
Black box testing , the tester is no Knowledge of its inner structure(i.e. how it woks)The tester only has knowledge of what it does(Focus only input & output)	The White box approach focuses on the inner structure of the software to be tested.
Black box approach is usually applied large size piece of software.	White box approach is usually applied small size piece of software.
Black box testing sometimes called functional or specification testing.	White box sometimes called clear or glass box testing.

4. what is a control flow graph? How is it used in white box test design?

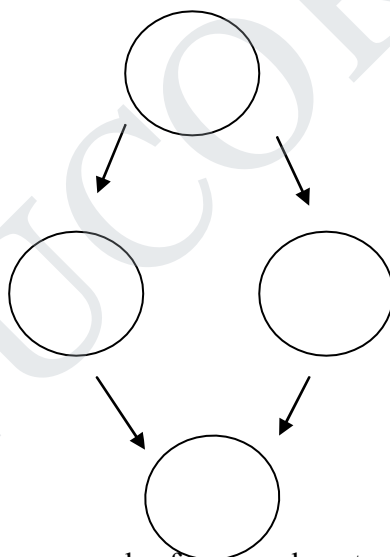
- i. Sequential ( e.g., Assignment statements)
- ii. Condition (e.g., if/then/else statements)
- iii. Iteration (e.g., while, for loops)

The graphical representation of these three primes are given

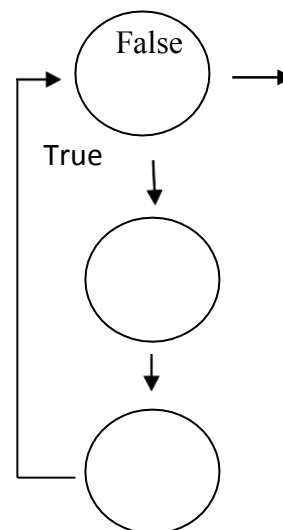
Sequence



Condition



Iteration



5. What inference can you make from random testing. Requirement based testing and domain testing. Explain?

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

Requirements-based testing is a testing approach in which test cases, conditions and data are derived from requirements. It includes functional tests and also non-functional attributes such as performance, reliability or usability.



## IT8076-SOFTWARE TESTING

---

Domain testing is a technique for testing software in which a minimum number of inputs are used to test the output of a system, to be sure that the system does not accept invalid and input values that are out of range. It is one of the essential white box testing methods.

6. Explain the differences between random testing and testing using error guessing.

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

The tester/developer is sometimes able to make an educated “guess” as to which type of defects may be present and design test cases to reveal them. Error Guessing is an ad-hoc approach to test design in most cases.

7. Explain about Boundary value analysis:

Boundary value analysis strengthen the test cases developed by equivalence class partitioning

Rules:

- i. Input condition is range of values, valid input at ends, invalid at one below and above
- ii. Number of values, valid is max, min value, invalid is is one less than and greater than max and min.
- iii. Input condition is set of values, valid is one in the set, invalid other than one in the set

---

## IT8076-SOFTWARE TESTING

---

### 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. Define Load.

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

#### 11. List the two major requirements of Performance testing.

- ✓ Functional requirements
- ✓ Quality requirements.

---

## IT8076-SOFTWARE TESTING

---

**12. List the several types of system tests.**

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

**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 no defect has been introduced due to the changes.

**19. List the effect of security breaches.**

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

**20. 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.

---

## IT8076-SOFTWARE TESTING

---

### PART- B

1. How would you define a software unit? In terms of your definition, what constitutes a unit for procedural code for object-oriented code?

- Unit: smallest possible testable component
- Procedures and functions
- Classes/ objects and methods
- Procedure sized – reusable components(COTS)

2. Discuss the issues that arise in class testing.

- Adequately testing classes
- Observation of object state and state changes
- Retesting of classes-I
- Retesting of classes-II

3. Why is it so important to design a test harness for reusability?

Auxiliary code developed into support 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.

4. What are the key differences in integrating procedural-oriented systems as compared to object-oriented systems?

OOP is a high-level programming language where a program is divided into small chunks called objects using the object-oriented model, hence the name. This paradigm is based on objects and classes.

- **Object** – An object is basically a self-contained entity that accumulates both data and procedures to manipulate the data. Objects are merely instances of classes.

- **Class** – A class, in simple terms, is a blueprint of an object which defines all the common properties of one or more objects that are associated with it. A class can be used to define multiple objects within a program.

- POP follows a step-by-step approach to break down a task into a collection of variables and routines (or subroutines) through a sequence of instructions. Each step is carried out in order in a systematic manner so that a computer can understand what to do. The program is divided into small parts called functions and then it follows a series of computational steps to be carried out in order.

- It follows a top-down approach to actually solve a problem, hence the name. Procedures

## IT8076-SOFTWARE TESTING

---

5. Describe the activities/Tasks and responsibilities for developer/testers in support of multilevel testing.

- Apply black and white box methods
- Assist with test planning
- Test at all levels
- Train and mentor
- Participate in task forces
- Interact with users

6. State and describe different levels of Testing.

- a. Unit Test
- b. Integration Test
- c. System Test
- d. Acceptance Test

7. How would you classify integration testing and system testing.

At the integration level several components are tested as a group and the tester investigates component interactions. When integration test are completed a software system has been assembled and its major subsystems have been tested. At this point the developers /testers begin to test it as a whole. System test planning should begin at the requirements phase.

8. Differentiate alpha testing from beta testing and discuss in detail about the phases in which alpha and beta testing is done. In what way it is related to milestone and deliverable.

- Alpha test developer's to use the software and note the problems.
- Beta test who use it under real world conditions and report the defect to the Developing organization.

---

## IT8076-SOFTWARE TESTING

---

### 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.

---

## IT8076-SOFTWARE TESTING

---

### 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 carrying 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.

---

## IT8076-SOFTWARE TESTING

---

### PART - B

1. Why is testing planning so important for developing a repeatable and managed testing process?

Business goal: To increase market share 10% in the next 2 years in the area of financial software

Technical Goal: To reduce defects by 2% per year over the next 3 years.

Business/technical Goal: To reduce hotline calls by 5% over the next 2 years

Political Goal: To increase the number of women and minorities in high management positions by 15% in the next 3 years.

2. . What role do managers play in support of a test group?

#### **Managers:**

Task forces, policies, standards, planning Resource allocation, support for education and training, Interact with users/Clients

3. Discuss in detail about the test specialist skills.

#### **Personal and managerial Skills**

Organizational, and planning skills, work with others, resolve conflicts, mentor and train others, written /oral communication skills, think creatively.

#### **Technical Skills**

General software engineering principles and practices, understanding of testing principles and practices, ability to plan, design, and execute test cases, knowledge of networks, database, and operating System.

4. Discuss in detail about the test plan components.

- Test plan identifier
- Introduction
- Items to be tested
- Features to be tested
- Approach
- Pass/fail criteria
- Suspension and resumption criteria
- Test deliverables
- Testing Tasks
- Test environment
- Responsibilities
- Staffing and training needs
- Scheduling



## IT8076-SOFTWARE TESTING

---

- Risks and contingencies
- Testing costs
- Approvals.

5. Describe with example test people management. How will you build a testing group, discuss with an example.

- Managers:

Task forces, policies, standards, planning Resource allocation, support for education and training, Interact with users/Clients

- Developers/Testers

Apply Black box and White box methods, test at all levels, Assst with test planning, Participate in task forces.

- Users/Clients

Specify requirement clearly, Support with operational profile, Participate in acceptance test planning

6. Develop the challenges and issues faced in testing service organization, also write how we can eliminate challenges.

- Complete testing is impossible. For any application, there are many different test combinations and resulting scenarios. ...

- Setting the right process. ...
- Lack of proper communication. ...
- Lack of resources. ...
- Test coverage.

---

## IT8076-SOFTWARE TESTING

---

### 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.

---

## IT8076-SOFTWARE TESTING

---

### 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

## IT8076-SOFTWARE TESTING

---

### 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

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

### PART-B

1. Discuss in detail about the controlling and monitoring: three critical views.

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

2. Explain in detail about the role of reviews in testing software deliverables.

- Identify problem components or components in the software artifact that need improvement.
- Identify components of the software artifact that donot need improvement.
- Identify specific errors or defects in the software artifact.
- Ensure that the artifact confirms to organizational standards.

3. Discuss in detail about the components of review plans.

- Review Goals
- Items being reviewed
- Preconditions for the review.
- Rolls,Team size,participants.
- Training requirements.
- Review steps.
- Time requirements

## IT8076-SOFTWARE TESTING

---

### 4. Explain about the various types of reviews.

**Inspection :** It is a type of review that is formal in nature and requires prereview preparation on the part of the review team. The Inspection leader prepares is the checklist of items that serves as the agenda for the review.

**WalkThroughs :** It is 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 .It have traditionally been applied to design and code.

### 5. Explain the design and architecture for automation.

- External models
- Scenario and configuration file modules
- Test case and test frame work module
- Tools and result modules
- Report generation and report/ metrics modules

### 6. List out the generic requirement for test tool/framework. Outline the challenges in automation.

- No hard coding in test suite
- Test case/ suite expandability
- Reuse of code for different types of testing
- Automatic set up and clean up
- Independent test cases
- Test case dependency
- Including test cases during execution
- Coding standards and directory structure
- Selective execution of test cases
- Random execution of test cases
- Parallel execution of test cases
- Looping the test cases
- Grouping test scenario
- Execution based on previous result
- Remote execution
- Automatic archival of test data
- Reporting scheme
- Independent of languages.