



"Techno - Social Excellence"
Marathwada Mitramandal's
Institute of Technology (MMIT)

Accredited with "A" Grade by NAAC

Survey No. 35, Vadgaon Shinde Road, Lohgaon, Pune - 411 047

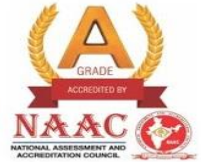
Approved by AICTE, New Delhi, Recognised by DTE, M.S.Mumbai, Affiliated to Savitribai Phule Pune University

Email : principal@mmit.edu.in

Website : www.mmit.edu.in

Tel No. : +91 7447786623 / +91 7447786624

DTE Institute Code : 6203



STQA MCQ QUESTIONS (50)

1.1 What is software testing?

- (a) It is the process of demonstrating that errors are not present.
- (b) It is the process of establishing confidence that a program does what it is supposed to do.
- (c) It is the process of executing a program with the intent of finding errors.**
- (d) It is the process of showing the correctness of a program.

1.2 Why should testing be done?

- (a) To ensure the correctness of a program
- (b) To find errors in a program**
- (c) To establish the reliability of a program
- (d) To certify the effectiveness of a program

1.3 Which phase consumes maximum effort to fix an error?

- (a) Requirements analysis and specifications
- (b) Design phase
- (c) Coding phase**
- (d) Feasibility study phase

1.4 Which objective is most difficult to achieve?

- (a) Execute every statement of a program at least once
- (b) Execute every branch statement of a program at least once
- (c) Execute every path of a program at least once**
- (d) Execute every condition of a branch statement of a program at least once

1.5 Software errors during coding are known as:

- (a) Bugs**
- (b) Defects
- (c) Failures
- (d) Mistakes

1.6 The cost of fixing an error is:

- (a) More in requirements analysis and specification phase than coding phase
- (b) More in coding phase than requirements analysis and specification phase**
- (c) Same in all phases of a software development life cycle
- (d) Negligible in all phases

1.7 Beta testing is done by:

- (a) Developers
- (b) Testers
- (c) Potential customers**
- (d) Requirements writers

1.8 Alpha testing is carried out at the:

- (a) Developer's site in a controlled environment**
- (b) Developer's site in a free environment
- (c) Customer's site in a controlled environment
- (d) Customer's site in a free environment

1.9 The purpose of acceptance testing is:

- (a) To perform testing from the business perspective**
- (b) To find faults in the software
- (c) To test the software with associated hardware
- (d) To perform feasibility study

1.10 Acceptance testing is done by:

- (a) Developers
- (b) Customers**
- (c) Testers
- (d) All of the above

1.11 Program is:

- (a) Subset of software**
- (b) Superset of software
- (c) Set of software
- (d) Union of software

1.12 Which is not an infrastructure software?

- (a) Compiler
- (b) Operating system
- (c) Testing tools
- (d) Result Management Software**

1.13 Software should have:

- (a) Program + operating system + compiler
- (b) Set of programs + operating system
- (c) Programs + documentation + operating procedures**
- (d) None of the above

1.14 Concepts of software testing are applicable to:

- (a) Procedural programming languages
- (b) Object oriented programming languages
- (c) 'C', 'C++' and Java programming languages
- (d) All of the above**

1.15 CASE Tool is:

- (a) Computer Aided Software Engineering Tool**
- (b) Component Aided Software Engineering Tool
- (c) Constructive Aided Software Engineering Tool
- (d) Complete Analysis Software Enterprise Tool

1.16 One fault may lead to:

- (a) One failure
- (b) Many failures
- (c) No failure
- (d) All of the above**

1.17 Test suite of a program is a:

- (a) Set of test cases**
- (b) Set of inputs with pre-conditions
- (c) Set of outputs with post-conditions
- (d) Set of testing strategies

1.18 Alpha and Beta testing techniques are related to:

- (a) Unit testing
- (b) Integration testing
- (c) System testing
- (d) Testing by Customer**

1.19 Testing a software is primarily focused on:

- (a) Verification activities only
- (b) Validation activities only
- (c) Verification and validation activities**
- (d) None of the above

1.20 Testing a software with real data in real environment is known as:

- (a) Alpha testing
- (b) Beta testing**
- (c) System testing
- (d) Integration testing

1.21 Verification activities are:

- (a) Performed manually
- (b) Related to reviewing the documents and source code
- (c) Known as static testing
- (d) All of the above**

1.22 Validation activities are:

- (a) Dynamic activities and require program execution**
- (b) Related to inspecting the source code
- (c) Related to static testing
- (d) Related to source code design and documentation

1.23 When the output of a program is different from the expected output, it is known as:

- (a) A fault

- (b) An error
- (c) A failure**
- (d) A mistake

1.24 Software testing activities should be started:

- (a) After the completion of source code
- (b) After the completion of design phase
- (c) As early as possible in the software development life cycle**
- (d) After the completion of software requirements and analysis phase

1.25 Software testing activities are important in:

- (a) Every phase of the software development life cycle
- (b) The last few phases of the software development life cycle
- (c) The software requirements and analysis phase
- (d) All of the above**

1.26 The focus of acceptance testing is:

- (a) To find faults
- (b) To ensure correctness of software
- (c) To test integration related issues
- (d) To test from the user's perspective**

1.27 A reliable software is one which is:

- (a) Liked by its users
- (b) Delivered on time and with budget
- (c) Unlikely to cause failures**
- (d) Very easy to use

1.28 When to stop testing and release the software to customers should be decided on the basis of:

- (a) Market conditions
- (b) Budget and availability of resources
- (c) Test metrics**
- (d) Capabilities of the testing persons

1.29 What are the good software testing practices?

- (a) Involve testing persons as early as possible in the software development life cycle
- (b) Apply effective verification techniques
- (c) Enforce inspections and reviews after every phase of the software development life cycle
- (d) All of the above**

1.30 What is a test case?

- (a) Input(s), expected output(s), pre-condition(s) and post-condition(s)**
- (b) Steps of execution
- (c) A list of activities which can be tested
- (d) None of the above

1.31 You cannot control what you cannot :

- (a) Define
- (b) Measure**

- (c) Improve
- (d) Change

1.32 What is the major benefit of verifications in the early phases of the software development life cycle?

- (a) It identifies changes in the SRS
- (b) It reduces defect multiplication**
- (c) It allows involvement of testing persons
- (d) It improves discipline in the various development activities

1.33 Behavioural specifications are required for:

- (a) Modelling**
- (b) Verification
- (c) Validation
- (d) Testing

1.34 Which, in general, is the least expected skill of a testing person?

- (a) Diplomatic**
- (b) Reliable
- (c) Having good attention to detail
- (d) Good developer

1.35 Debugging of a program is

- (a) The process of executing the program
- (b) The process of identifying a fault and removing it from the program**
- (c) The process of experiencing a failure
- (d) The process of improving the quality of the program

1.36 All validation activities come under the category of:

- (a) Dynamic testing**
- (b) Static testing
- (c) Source code design
- (d) None of the above

1.37 All verification activities come under the category of:

- (a) Dynamic testing
- (b) Static testing**
- (c) Source code design
- (d) None of the above

1.38 Which is not a factor of software quality?

- (a) Reliability
- (b) Portability
- (c) Efficiency
- (d) Functionality**

1.39 Which is the most important factor of software quality?

- (a) Reliability**
- (b) Understandability

- (c) Efficiency
- (d) Consistency

1.40 Quality assurance activities concentrate on

- (a) Software design
- (b) Software performance
- (c) Software products
- (d) Software processes**

1.41 Which is not the quality of a testing person?

- (a) Cautious
- (b) Curious
- (c) Judgmental**
- (d) Critical

1.42 What should be the best possible objective for testing?

- (a) Execute every statement at least once
- (b) Execute every path at least once**
- (c) Execute every branch statement at least once
- (d) Execute every condition of a branch statement at least once

1.43 Which is not a user manual?

- (a) Reference guide
- (b) Beginner's guide
- (c) Sequence diagrams**
- (d) System overview

1.44 Which is not a documentation manual?

- (a) SRS document
- (b) SDD document
- (c) Source code
- (d) Installation guide**

1.45 Which is not the limitation of testing?

- (a) Difficult to measure the progress of testing
- (b) Availability of testing tools**
- (c) Input domain is too large to test
- (d) Too many paths in the program

1.46 How much percentage of cost is generally consumed in software testing with reference to software development cost?

- (a) 10 – 20
- (b) 40 – 50**
- (c) 80 – 90
- (d) 70 – 80

1.47 How much testing is enough?

- (a) Not easy to decide**
- (b) Depends on complexity and criticality

- (c) Depends on abilities of testing persons
- (d) Depends on maturity of developers

1.48 If an expected output is not specified then:

- (a) We cannot execute the test case
- (b) We may not be able to repeat the test
- (c) We may not be able to decide if the test has passed or failed**
- (d) We may not be able to automate the testing activity

1.49 Which of the following is a reason for a software failure?

- (a) Testing fault
 - (b) Software Fault**
 - (c) Design Fault
 - (d) Requirement Fault
- 34 Software Testing

1.50 Why is it impossible to test a program completely?

- (a) Input domain is too large to test**
- (b) Good testers are not available
- (c) Efficient testing tools are not available
- (d) None of the above