FAQ on Testing Concepts

Q. 1	**What is Exploratory Testing and when should it be performed?**
Ans.	• simultaneous test design and execution” against an application. • Usually performed as a final check before the software is released.
Q. 2	**Why to use test design techniques?**
Ans.	• To help identify defects • To reduce the number of test cases • To achieve 100% test Coverage with minimum no of Test cases
Q. 3	**What is the difference between Re-testing and Regression testing?**
Ans.	**Re-testing (Confirmation Testing):** Re-execution of the test after defects are fixed. • The test is executed in exactly the same way as it was the first time • Same environment, versions • Same inputs and preconditions **Regression Testing:** Ensures that the software is not adversely affected by the changes and critical functionality of the software is still intact. • To look for any unexpected side-effects • After software changes, including faults fixed • When the environment changes, even if application stays same • For emergency fixes (possibly a subset)
Q. 4	**What is Verification?**
Ans.	Verification refers to a set of activities which ensures that software correctly implements a specific function
Q. 5	**What is Validation?**
Ans.	Validation refers to a different set of activities which ensures that the software that has been built is traceable to customer requirements
Q. 6	**Verification VS** **Validation**
Ans.	1.Are we building the product, right? Are we building the right product? 2.Process oriented Product oriented 3.Static testing Dynamic testing 4.Conducted by QA team QC team 5.low level activity High level activity 6.Review (Walkthrough, Inspections etc.) activity Unit testing, Integration testing System Testing etc. 7.preventing action correcting action
Q. 7	**What are the 7 principles of Testing?**
Ans.	• Principle 1 - Testing shows presence of defects but cannot prove that there are no defects • Principle 2 - Exhaustive testing is impossible • Principle 3 - Early testing • Principle 4 - Defect Clustering • Principle 5 - Pesticide Paradox • Principle 6 - Testing is context dependent • Principle 7 - Absence of Errors fallacy
Q. 8	**difference between Load and stress Testing?**
Ans.	**Example: Expected users are 1K who can access the application** Ensure that a program never returns inaccurate result even though valid data is passed. • **Load**: Testing an app with no of expected users (i.e 1K) • **Stress**: Testing an app beyond the no of expected users (i.e <1K)
Q. 9	**What is the difference between static and dynamic testing?**
Ans.	• **Static testing:** During Static testing method, the code is not executed, and it is performed using the software documentation.

	• **Dynamic testing:** To perform this testing the code is required to be in an executable form**.**
Q. 10	**Importance of Software Testing**
Ans.	• Ensures that Customer's Objectives are met • Early detection of errors to prevents breakdown at a later stage • Ensures that the software is reliable and usable • Ensures effective execution in the given environment • Reduces overall cost of software
Q. 11	**Define the terms Error, Fault, Bug, Failure and Defect.**
Ans.	• **Error(Mistake):** A human action that produces an incorrect result • **Fault:** A stage caused by an error which leads to unintended functionality of the program • **Bug:** It is an evidence of the fault. It causes the program to perform in unintended manner. It is found before application goes into beta version • **Failure:** Inability of the system to perform functionality according to its requirement • **Defect:** It is a mismatch of the actual and expected result identified while testing the software in the beta version
Q. 12	**What is the definition and purpose of testing and debugging?**
Ans.	▪ Testing is the process of executing a program with the intent of finding errors ▪ Testing is a process used to help identify the correctness, completeness and quality of a developed computer software ▪ Debugging is an art used to "isolate", and "correct" the cause of an error ▪ Debugging is performed by developers to uncover where a defect in the code exists and correct it
Q. 13	**What is static and dynamic testing? What are the techniques of static and dynamic testing?**
Ans.	**Static Testing:** Testing a software without execution on a computer • Review : Review the created artifacts using checklist • Code Inspection : Code inspection is a set of procedures and error detection techniques for group code reading. • Walkthrough : Like code inspection it is also an group activity. **Dynamic Testing** : Testing a software by execution using sample input values. • White Box testing : Used to test the internal structure of the code • Black Box Testing : Test the functionality of application by providing input and getting expected output
Q. 14	**What are the guidelines for implementing test cases**
Ans.	• Write test case for all the requirements specified in the application • Take care of writing test case for non-functional requirements like security, performance, etc.. • If any test case fails, log the failed test cases as defect in defect tracking sheet. • Check for all boundary conditions. • **80-20 Rule:** In most systems, 20% of the modules account for 80% of the defects found. The probability of finding defect in a module is directly proportional to the number of defects already found in the module. • Do self-review and peer review for all test cases as quality of test case affects testing.
Q. 15	**What is system testing, validation testing, acceptance and regression testing?**
Ans.	**System Testing :** a complete integrated system as a whole, in order to evaluate compliance with respect to specified requirements **Validation Testing :** It checks whether the program matches it's external specifications and to have a final check to see whether it is indeed the right product **Acceptance Testing :** Acceptance Testing focuses on testing whether the right system has been created. It is usually carried out by the end user **Regression Testing :** Regression Testing involves "selective re-testing" of the system or it's components after the changes are done. It is done to verify absence of unintended effects and to verify compliance with all (old and new) requirements.
Q. 16	**What is Risk Based Testing?**
Ans.	**Risk based Testing is used to reduce risk of adverse effect occurring or to reduce the impact of adverse effect**

	▪ A factor that could result in negative consequences; usually expressed as impact and like hood ▪ Risks are used to decide where to start testing and where to test more.

Q. 17	**What is Project and Product Risk?**
Ans.	**Project Risk** – A risk related to management and control of the (test) project is called as Project Risk. ▪ Organizational factor ▪ Technical issues ▪ Supplier issues **Product Risk** -- it is directly related to the test object. (Risks related to quality of a product) ▪ Failure-prone software delivered ▪ Poor software characteristics ▪ Poor data integrity and quality
Q. 18	**What is a need of Independent Testing?**
Ans.	It is conducted by an independent test team other than developer to avoid author bias and is more effective in finding defects and failures ▪ The tester sees what has been built rather than what the developer thought ▪ Unbiased testing is necessary to objectively evaluate quality of a software ▪ Developer carrying out testing would not like to expose defects ▪ The tester is totally unbiased
Q.19	**What is White Box Testing? And What are the types of White Box Test design techniques?**
Ans.	**White Box Testing:** (structural, glass-box and clear-box testing) ▪ To examine the internal structure of the program ▪ It makes sure that each statement, decision branch, or path is tested with at least one test case **Types of White Box Test Design techniques:** ▪ Code Coverage • Statement Coverage • Decision Coverage • Condition Coverage • Loop Testing ▪ Code complexity • Cyclomatic Complexity ▪ Memory Leakage
Q.20	**What is Black Box Testing? And What are the types of Black Box Test design techniques?**
Ans.	**Black Box Testing:** (behavioral, functional, opaque-box and closed-box testing) ▪ Black box is data-driven, or input/output-driven testing ▪ The Test Engineer is completely unconcerned about the internal behavior and structure of program **Types of Black Box Test design techniques:** ▪ Equivalence Partitioning ▪ Boundary Value Analysis ▪ Use Case / UML ▪ Error Guessing ▪ Cause-Effect Graphing ▪ State Transition Testing
Q.21	**What is Test Case, Test Suite and Test Cycle?**

Ans.

- **Test Case** - A set of inputs, execution preconditions, and expected outcomes developed for an objective, such as to exercise a program path or to verify compliance with a specific requirement.
- **Test Suite** – A set of individual test cases/scenarios that are executed as a package, in a sequence.
- **Test Cycle** – A test cycle consists of a series of test suites which comprises a complete execution set from the initial setup to the test environment through reporting and clean up.

Q.22 What are the properties of good test data?

Ans.

- **Realistic** – accurate in context of real life
  E.g. Age of a student giving graduation exam is at least 18
- **Practically valid** – data related to business logic
  E.g. Age of a student giving graduation exam is at least 18 says that 60 years is also valid input but practically the age of a graduate student cannot be 60
- **Cover varied scenarios**
  E.g. Don't just consider the scenario of only regular students but also consider the irregular students, also the students who are giving a re-attempt, etc.
- **Exceptional data**
  E.g. There may be few students who are physically handicapped must also be considered for attempting the exam

Q.23 What is Positive and Negative Testing?

Ans.

Positive testing: (Test to Pass, Happy Path Testing)

- It can be performed on the system by entering the valid data as input
- It is generally the first form of testing that a tester performs on an application

Negative Testing: (Test to Fail)

- To break the system and to verify the application response for invalid inputs

Q.24 What is Basic and Alternate Testing?

Ans.

Basic Test:

- Used to test very basic functionality of software
  E.g : In Calculator check the output for 2+2=4 or not
- Basic test are always positive tests
- Basic test can be smoke test or sanity test

Alternate Test:

- performed to meet its requirements but using different route than the obvious/basic path
- Alternate test is a kind of positive testing

Q.25 What are the parallel phase of V Model?

Ans.

Requirement Specification	Acceptance Test Cases
Functional Design	System Test Cases
Detailed Design	Integration Test Cases
Program Specification	Unit Test Cases

Q.26 What are the phase of RUP (Rational Unified Process) Model?

Ans.

Inception: Define scope of system

Elaboration: Mitigate the risk items and defines the architecture of the project

Construction: Build the software system, Development of component

Transition: Transit the system from development into production. Beta testing is performed**.**

Q.27 What are the phases of Testing?

Ans.

Unit (Component) testing:

- Unit testing is code-based and performed primarily by developers

Integration testing:

- Integration testing demonstrates that two or more units work together properly

System testing:

- System testing demonstrates that the system works end-to-end in a production-like environment to provide the business functions specified in the high-level design.

Acceptance testing:

- Acceptance testing is conducted by business owners and users to confirm that the system does, in fact, meet their business requirements.

Q.28 **What are the types of Integration Testing and their approaches?**

Ans.

1. **Incremental Integration Testing**
   - Big Bang Approach
2. **Non-Incremental Integration Testing**
   - Top Down Approach
   - Bottom Up Approach

Q.29 **What is Top Down Integration testing?**

Ans. **Top Down Approach:** Top module is tested first. Once testing of top module is done then any one of the next level modules is added and tested. This continues till last module at lowest level is tested

- **Stubs** are substituted for all components directly subordinate to the main control module
- Depending on the approach subordinate stubs are replaced by actual components

Q.30 **What is Bottom Up Integration testing?**

Ans. **Bottom Up Approach:** Module at the lowest level is tested first. Once testing of that module is done then any one of the next level modules is added to it and tested. This continues till top most module is added to rest all and tested

- A **driver** is written to coordinate test case input and output
- Drivers are removed, and clusters are combined moving upward in the program structure

Q.31 **What is System Testing? What are the different types of system testing?**

Ans. System Testing is End to End Testing.

We perform Functional and Non-functional Testing in System testing**.**

Types of System Testing:

1. **Non-Functional Testing**
   - Performance Testing
   - Localization Testing
   - Usability Testing
   - Recovery Testing
   - Documentation Testing
   - Configuration Testing

   Web Security Testing

   Volume Testing

   Load Testing

   Stress Testing

   Security Testing

   Installation Testing
2. Functional Testing
3. User Acceptance Testing
4. Testing related to Changes: Re-Testing and Regression Testing
5. Re-testing (Confirmation Testing)
6. Regression Testing
7. Exploratory Testing
8. Maintenance Testing

Q.32 What is Security Testing and Web Security Testing?

Ans. Security Testing verifies that protection mechanisms built into the system will protect it from improper penetration.

- E.g : One tries to subvert the DBMS's data security mechanisms

Web application security is a branch of Information Security that deals specifically with security of web applications.

- E.g: Phishing attacks on banking sites

Q.33	What is Installation Testing and Configuration Testing?
Ans.	Installation testing: Installer is the first contact a user has with a new software!!! ▪ It is to ensure Application is getting installed properly ▪ New program that is installed is working as desired ▪ Old programs are not hampered Configuration testing: Analyse system behaviour in various hardware and software configurations specified in the requirements.
Q.34	**What is Localization Testing, Documentation Testing and Recovery Testing?**
Ans.	**Localization Testing:** Localization translates the product UI and occasionally changes some settings to make it suitable for another region. ▪ Culture/locale-specific, language specific and region-specific areas **Documentation Testing:** This testing is done to ensure the validity and usability of the documentation ▪ This testing is done to ensure the validity and usability of the documentation **Recovery Testing:** This test confirms that the program recovers from expected or unexpected events. Events can include shortage of disk space, unexpected loss of communication ▪ To check recovery is automatic, data recovery and restarts are evaluated for correctness
Q.35	**What is Usability Testing?**
Ans.	How much User-friendly application is!!! ▪ Effective–- Accomplishes user’s goal ▪ Efficient-- Accomplishes the goal quickly ▪ Satisfaction–- User enjoys the experience To Check Layouts, Readability, Display characteristics, Navigations and Time sensitivity.
Q.36	**What is UAT (user acceptance Testing)?**
Ans.	A test executed by the end user(s) in an environment simulating the operational environment. ▪ Not a responsibility of the Developing Organization ▪ Usually carried out by the end user Two types of UAT: ▪ Alfa Testing: Performed at the developer's site by a cross-section of potential users ▪ Beta Testing: Performed by a cross-section of users who install it and use it under real-world working conditions
Q.37	What is exploratory testing?
Ans.	Exploratory testing is simultaneous learning, test design, and test execution ▪ Also known as “Random” testing or “Ad-hoc” testing ▪ Careful Observation ▪ Critical thinking ▪ Diverse Ideas ▪ Pooling resources (knowledge, learnings)
Q.38	What is Maintenance Testing?
Ans.	Testing done after the system is deployed or on existing system ▪ two parts: 1. Testing the changes and defects 2. Regression tests ▪ Impact and risk analysis is important activity performed to determine Test efforts ▪ It is triggered by planned modifications, Ad-hoc corrective modifications, migration, or retirement of the system

Q.39	What is Test Plan and what are the Test Planning activities?
Ans.	Test Plan: A document Describing the Scope, approach, resources and schedule of the intended test activities. ▪ The Test plan itself serve as vehicles for communicating with project team members ▪ The Test plan helps us manage changes ▪ There can be different Test plans for different Test levels Test Plan Activities: ▪ Test Items ▪ Features to be Tested ▪ Features not to be Tested ▪ Test Approach (Strategy) ▪ Item Pass/Fail Criteria ▪ Test Deliverables ▪ Environmental Needs ▪ Responsibilities ▪ Schedule etc.
Q.40	What is Entry and Exit criteria?
Ans.	Entry Criteria: When to start testing Exit Criteria: When to stop testing
Q.41	What are the test case execution preconditions?
Ans.	▪ Setting up the Environment: Hardware, software, access to application etc. ▪ Setting up data for Execution: create fresh data, use existing sample, verify data is not corrupted ▪ Design test data like no data, valid/in-valid data set, boundary data set etc.
Q.42	What are the types of test environment?
Ans.	▪ Unit Test Environment ▪ Assembly/Integration Test Environment ▪ System/Functional/QA Test Environment ▪ User Acceptance Test Environment ▪ Production Environment
Q.43	What is Test Metrics and what are the different types of Test Metrics?
Ans.	Metrics should be collected during and at the end of a test level. They are also valuable input into process improvement Types of Test Metrics: ▪ Project Metrics ▪ Process Metrics ▪ Productivity Metrics ▪ Closure Metrics
Q.44	What are the tools support for Management of testing?
Ans.	▪ Test Management Tools ▪ Requirements Management Tools ▪ Incident Management Tools (Defect Tracking Tools) ▪ Configuration Management Tools
Q.45	What are the tools support for Static testing and test specification?
Ans.	Static Testing tools: ▪ Review Tools ▪ Static Analysis Tools ▪ Modeling Tools Test Specification tools: ▪ Test Design Tools ▪ Test Data Preparation Tools

Q.46	What are the tools support for Test Execution and Logging?
Ans.	▪ Test Execution Tools ▪ Test Harness/Unit Test Framework Tools ▪ Test Comparators ▪ Coverage Measurement Tools ▪ Security Testing Tools
Q.47	▪ List out some benefits and risk of using tools.
Ans.	Benefits: ▪ Reduction of repetitive work ▪ Greater consistency and repeatability ▪ Objective assessment ▪ Ease of access to information about tests or testing Risk: ▪ Unrealistic expectations from the tool ▪ Under estimating the time, cost and effort while initial introduction of a tool ▪ Poor response from vendor for support, upgrades and defect fixes ▪ Risk of suspension of open-source / free tool project
Q.48	What is configuration Management?
Ans.	A discipline applying technical and administrative direction and surveillance to identify and document the functional and physical characteristics of a configuration item
Q.49	What is Configuration Control or Version control?
Ans.	An element of configuration management, consisting of evaluation, coordination, approval or disapproval and implementation of changes to configuration items after formal establishment of their configuration identification
Q.50	What is Test Control?
Ans.	Test control is the response to Test Monitoring and Test Reporting that allows us to be IN CONTROL of the project E.g: ▪ Re-prioritize tests when an identified risk ▪ Change the test schedule based on availability of a test environment