1. **What is software testing?**

Ans. Software Testing is a process used to identify the correctness, completeness, and

quality of developed computer software. Software testing is a process of executing a program or application with the intent of finding

the software bugs.The process consisting of all life cycle activities, both static and dynamic, concerned

with planning, preparation and evaluation of software products and related work

products to determine that they satisfy specified requirements, to demonstrate that

they are fit for purpose and to detect defects.

1. **What is Exploratory Testing?**

Ans. Exploratory testing is a concurrent process where Test design, execution and logging happen simultaneously,Testing is often not recorded, Makes use of experience, heuristics and test patterns, Testing is based on a test charter that may include

Scope of the testing (in and out).The focus of exploratory testing is more on testing as a "thinking" activity.

1. **What is traceability matrix?**

Ans. To protect against changes you should be able to trace back from every system component to the original requirement that caused its presence. A software process should help you keeping the virtual table up-to-date. Traceability Matrix (also known as Requirement Traceability Matrix - RTM) is a table

which is used to trace the requirements during the Software development life Cycle. It can be used for forward tracing (i.e. from Requirements to Design or Coding) or backward (i.e. from Coding to Requirements). There are many user defined templates for RTM.

1. **What is Boundary value testing?**

Ans. Software testing technique in which tests are designed to include representatives of boundary values. It is performed by the QA testing teams.

1. **What is Equivalence partitioning testing?**

Ans. Software testing technique that divides the input data of a software unit into partitions of data from which test cases can be derived. it is usually performed by the QA teams.

1. **What is Integration testing?**

Ans. The phase in software testing in which individual software modules are combined and tested as a group. It is usually conducted by testing teams.

1. **What determines the level of risk?**

Ans. A properly designed test that passes, reduces the overall level of Risk in a system. Risk – ‘A factor that could result in future negative consequences; usually expressed as impact and likelihood’. Risk could be any future event with a negative consequence .You need to identify the risks associated with your project.

Risks are of two types, (1) Project Risks (2) Product Risk.

1. **What is Alpha testing?**

Ans. Alpha Testing is definitely performed and carried out at the developing organizations location with the involvement of developers.Type of testing a software product or system conducted at the developer's site. Usually it is performed by the end user. Alpha Testing is always performed at the time of Acceptance Testing when developers test

the product and project to check whether it meets the user requirements or not. It comes under the category of both White Box Testing and Black Box Testing. It is always performed in Virtual Environment.

1. **What is beta testing?**

Ans. Beta Testing (field testing) is performed and carried out by users or you can say people at their own locations and site using customer data. Beta Testing is always performed at the time when software product and project are marketed. It is performed in Real Time Environment. It is only a kind of Black Box Testing. Beta testing can be considered “pre-release” testing.

1. **What is component testing?**

Ans. Component (Unit) – A minimal software item that can be tested in isolation. It means “A unit is the smallest testable part of software.” Component Testing – The testing of individual software components. Unit Testing is a level of the software testing process where individual units/components of a software/system are tested. The purpose is to validate that each unit of the software performs as designed.

Sometimes known as Unit Testing, Module Testing or Program Testing.

1. **What is functional system testing?**

Ans. Functional System Testing : A requirement that specifies a function that a system or system component must perform. A Requirement may exist as a text document and/or a model. Accuracy,Interoperability,Compliance, Auditability, Suitability.

1. **What is Non-Functional Testing?**

Ans. Testing of those requirements that do not relate to functionality. Emphasis on non-functional requirements : Performance, Load, Data volumes, Storage, Recovery, Usability, Stress, Security.

1. **What is GUI Testing?**

Ans. The process of testing a product that uses a graphical user interface, to ensure it meets its written specifications. This is normally done by the testing teams. Graphical User Interface (GUI) testing is the process of testing the system’s GUI of the System under Test. GUI testing involves checking the screens with the controls like menus, buttons, icons, and all types of bars – tool bar, menu bar, dialog boxes and windows etc.

1. **What is Adhoc testing?**

Ans. Testing performed without planning and documentation - the tester tries to 'break' the system by randomly trying the system's functionality. It is performed by the testing teams. Adhoc testing can be achieved with the testing technique called Error Guessing.

1. **What is white box testing and list the types of white box testing?**

Ans. Testing based on an analysis of the internal structure of the component or system. Structure-based testing technique is also known as ‘white-box’ or ‘glass-box’ testing technique because here the testers require knowledge of how the software is implemented, how it works. The different types of coverage are: (1) Statement/Segment Coverage (2) Decision/Branch Coverage (3) Condition coverage.

1. **What is black box testing? What are the different black box testing techniques?**

Ans. Testing, either functional or non-functional, without reference to the internal structure of the component or system. Specification-based testing technique is also known as ‘black-box’ or input/output driven testing techniques because they view the software as a black-box with inputs and outputs. There are four specification-based or black-box technique : (1) Equivalence partitioning (2) Boundary value analysis (3) Decision tables (4) State transition testing (5) Use-case Testing (6) Other Black Box Testing (Syntax or Pattern Testing).

1. **Mention what are the categories of defects?**

Ans. 1) DataQuality/DatabaseDefects

2) CriticalFunctionalityDefects

3) FunctionalityDefects

4) Security Defects

5) UserInterfaceDefects

1. **Mention what bigbang testing is?**

Ans.BigBang integration testing all components or modules is integrated simultaneously,after which every thing istested as a whole. BigBang testing has the advantage that everything is finished before integration testing starts. The major disadvantage is that in general it is time consumingand difficult to trace the cause of failures be cause of this late integration.

1. **What is the purpose of exit criteria?**

Ans. Ans. In Thoroughnessmeasures,suchascoverage of requirements ofcodeorriskcoverage.

Estimates of defect densityor reliability measures.(E.g.how many defects open by category)

How do we know when to stoptesting?

Run out of time?

Run out of budget?

The business tells you it went live last night!

Boss says stop?

All defects have been fixed?

When out exit criteria have been met?

Purpose of exit criteria is to define when we STOP testing eitheratthe:

End of all testing–i.e.productGoLive

End of phase of testing(e.g.handoverfromSystemTesttoUAT)

1. **When should "Regression Testing" be performed?**

Ans. When the system is stable and the system or the environment

changes when testing bug fix releases as part of the maintenance phase It should be applied at all Test Levels. It should be considered complete when agreed completion criteria for regression testing has been met Regression test suites evolve overtime and given that they are run frequently are ideal candidates for automation.

1. **What is 7 key principles? Explain in detail?**

Ans. Testing shows presence of Defects - Testing can show that defects are present, but cannot prove that there are no defects. Testing reduces the probability of undiscovered defects remaining in the software but, even if no defects are found, it is not a proof of correctness.

Exhaustive Testing is Impossible! - Testing everything including all combinations of inputs and preconditions is not So, instead of doing the exhaustive testing we can use risks and priorities to focus testing efforts.possible.

Why do not Testing Everything? - Exhaustive testing of complex software applications: requires enormous resources, is too expensive, takes too long.

Early Testing - Testing activities should start as early as possible in the software or system development life cycle, and should be focused on defined objectives. Testing activities should start as early as possible in the development life cycle.

Defect Clustering - A small number of modules contain most of the defects discovered during pre-release testing, or are responsible for the most operational failures

Pesticide Paradox - If the same tests are repeated over and over again, eventually the same set of test cases will no longer find any new defects. To overcome this “pesticide paradox”, the test cases need to be regularly reviewed and

revised, and new and different tests need to be written to exercise different parts of the software or system to potentially find more defects.

Testing is Context Dependent - Testing is done differently in different contexts. Different kinds of sites are tested differently.

Absence of Errors Fallacy - If the system built is unusable and does not fulfill the user’s needs and expectations then finding and fixing defects does not help. Even after defects have been resolved it may still be unusable and/or does not fulfil the

users’.

1. **Difference between QA v/s QC v/s Tester?**

Ans. Quality Assurance

1) Activities which ensure the implementation of processes, procedures and standards in context to verification of developed software and intended requirements.

2) Focuses on processes and procedures rather than conducting actual testing on the system.

3) Process oriented activities.

4) Preventive activities

5) It is a subset of Software Test Life Cycle (STLC).

Quality Control

1) Activities which ensure the verification of developed software with respect to documented (or not in some cases) requirements.

2) Focuses on actual testing by executing Software with intend to identify bug/defect through implementation of procedures and process.

3) Product oriented activities.

4) It is a corrective process

5) QC can be considered as the subset of Quality Assurance.

Tester

1) Activities which ensure the identification of bugs/error/defects in the Software.

2) Focuses on actual testing.

3) Product oriented activities.

4) It is a preventive process.

5) Testing is the subset of Quality Control.

1. **Difference between Smoke and Sanity?**

Ans. Smoke Testing

1) Smoke Testing is performed to ascertain that the critical functionalities of the program is working fine.

2) The objective of this testing is to verify the "stability" of the system in order to proceed with more rigorous testing.

3) This testing is performed by the developers or testers.

4) Smoke testing is usually documented or scripted.

5) Smoke testing is a subset of Regression testing.

6) Smoke testing exercises the entire system from end to end.

7) Smoke testing is like General Health Check Up.

Sanity Testing

1) Sanity Testing is done to check the new functionality / bugs have been fixed.

2) The objective of the testing is to verify the "rationality" of the system in order to proceed with more rigorous testing.

3) Sanity testing is usually performed by testers.

4) Sanity testing is usually not documented and is unscripted.

5) Sanity testing is a subset of Acceptance testing.

6) Sanity testing exercises only the particular component of the entire system.

7) Sanity Testing is like specialized health check up.

1. **Difference between verification and Validation?**

Ans. Verification

1) The process of evaluating work-products (not the actual final product) of a development phase to determine whether they meet the specified requirements for that phase.

2) To ensure that the product is being built according to the requirements and design specifications. In other words, to ensure that work products meet their specified requirements.

3) Are we building the product right?

4) Plans, Requirement Specs, Design Specs, Code, Test Cases.

5) Reviews, Walkthroughs,Inspections .

Validation

1) The process of evaluating software during or at the end of the development process to determine whether it satisfies specified business requirements.

2) To ensure that the product actually meets the user’s needs, and that the specifications were correct in the first place. In other words, to demonstrate that the product fulfills its intended use when placed in its intended environment.

3) Are we building the right product?

4) The actual product/software.

5) Testing.

1. **Explain types of Performance testing.**

Ans. 1) Load testing

2) Stress testing

3) Endurance testing

4) Spike testing

5) Volume testing

6) Scalability testing

1. **What is Error, Defect, Bug and failure?**

Ans. Error: A discrepancy between a computed, observed, or measured value or condition and

the true, specified, or theoretically correct value or condition. This can be a

misunderstanding of the internal state of the software, an oversight in terms of memory

management, confusion about the proper way to calculate a value, etc.

Failure: The inability of a system or component to perform its required functions within

specified performance requirements. See: bug, crash, exception, and fault.

Bug: A fault in a program which causes the program to perform in an unintended or

unanticipated manner. See: anomaly, defect, error, exception, and fault. Bug is terminology

of Tester.

Defect: Commonly refers to several troubles with the software products, with its external

behavior or with its internal features.

1. **Difference between Priority and Severity**

Ans. Priority is Relative and Business-Focused. Priority defines the order in which we should

resolve a defect. Should we fix it now, or can it wait? This priority status is set by the tester

to the developer mentioning the time frame to fix the defect. If high priority is mentioned

then the developer has to fix it at the earliest. The priority status is set based on the customer

requirements.

Severity is absolute and Customer-Focused. It is the extent to which the defect can affect

the software. In other words it defines the impact that a given defect has on the system.

1. **What is Bug Life Cycle?**

Ans. Bug life cycle is nothing but the various phases a bug under goes after it is raised or

reported. The different phases of Bug life cycle are,

- New or Opened

- Assigned

- Fixed

- Tested

- Closed

1. **Explain the difference between Functional testing and NonFunctional testing**

Ans. Functional Testing

1) Functional testing is performed using the functional specification provided by the client and verifies the system against the functional requirements.

2) Functional testing is executed first.

3) Manual testing or automation tools can be used for functional testing.

4) Business requirements are the inputs to functional testing.

5) Functional testing describes what the product does.

6) Easy to do manual testing.

7) Types of Functional testing are

(a) Unit Testing

(b) Smoke Testing

(c) Sanity Testing

(d) Integration Testing

(e) White box testing

(f) Black Box testing

(g) User Acceptance testing

(h) Regression Testing

Non-Functional Testing

1) Non-Functional testing checks the Performance, reliability, scalability and other non-functional aspects of the software system.

2) Non functional testing should be performed after functional testing.

3) Using tools will be effective for this testing.

4) Performance parameters like speed , scalability are inputs to non-functional testing.

5) Nonfunctional testing describes how good the product works.

6) Tough to do manual testing.

7) Types of Nonfunctional testing are

(a) Performance Testing

(b) Load Testing

(c) Volume Testing

(d) Stress Testing

(e) Security Testing

(f) Installation Testing

(g) Penetration Testing

(h) Compatibility Testing

(i) Migration Testing