

# Ranga Reddy sir MANUAL TESTING INTERVIEW QUESTIONS

## **1). Why testing is necessary?**

It helps avoid the extra costs that occur to fix issues after the product is released to the market.

There are a bunch of situations in which the information and details of the users are stolen and they are used for the benefits. It is considered to be the reason why people look for tested and reliable products.

The reason why software testing is important is because it makes sure that the software is user-friendly and as per the user expectations

If the performance of the software is low, you will find that it brings your reputation down in the market. Users are not going to trust any people that's why testing is important.

## **2). What are the drawbacks of manual Testing?**

It uses a comparatively large number of human resources

It's costly in the short run. Ideally, though, the business is investing significant funds at the start to head off more prominent and more expensive headaches after the product is released

It is very time-consuming

The tester develops test cases based on their skills and previous experiences. There is no proof that they have covered all functions.

Testers cannot use a given test case again. Testers need to develop separate test cases for each new application

It does not cover all aspects of testing.

## **3). What is a Defect/Bug?**

Developers and programmers sometimes make mistakes which create bugs called defects. Most bugs come from mistakes that developers or programmer make. Software bugs are classified into three types:

- ✓ Software Defects by its Nature
- ✓ Software Defects by its Priority
- ✓ Software Defects by its Severity.

## **4).What is Failure?**

When certain conditions arise during testing, and the tester takes action to fix these flaws, the test results in a failure, also known as a software malfunction.

We can classify risks into following categories:

Product risk (factors relating to what is produced by the work, i.e. the thing we are testing).

Project risk (factors relating to the way the work is carried out, i.e. the test project)

## **5).Why software Application will have defects?**

### **Lack of Collaboration**

Business requirements may not reach the development/testing team sometimes due to a lack of communication.

### **Lack of Code Coverage**

Some software defects can occur due to poor Code Coverage. This can be caused if the code base is not maintained properly to support code scalability or the dead, unused codes are not discovered due to bad code writing.

### **Poor Test Coverage**

Poor Test coverage is often a result of very few tests written for each user story that lacks visibility into the code that was changed unexpectedly. Also, an unclear understanding of the features being delivered in the release leads to Inadequate Testing.

### **Choosing a wrong Testing Framework**

A testing framework provides an execution environment for automating test scripts. An inappropriate Testing Framework leads to defects in the software. Often test data present in one framework is different than another framework. Therefore, some of the defects go undetected.

## **6).What is verification and validation?**

### **Verification:**

Verification is the process of evaluating work-products of a development phase to determine whether they meet the specified requirements.

Verification testing includes different activities such as business requirements, system requirements, design review, and code walkthrough while developing a product.

### **Validation:**

Validation testing is testing where tester performed functional and non-functional testing.

Validation differs from verification testing, another important phase of the product development process. Verification testing is the process of confirming that the way a product performs meets the predetermined product specifications.

Developers can perform this kind of testing throughout the development process. After verifying that the final product meets the design specifications, the team can move on to the validation process to ensure those specifications can meet user needs.

## **7)What is the day to day job of a Tester?**

### **Testing**

The first task on the daily schedule of a software developer is the testing of software that he/she receives from the development team.

Testing includes checking the software for any bugs that may hamper its performance. Once the tester identifies a glitch, it is his/ her responsibility to track its origin and fix the bug.

### **Coordination with developers**

It is no surprise that communication and coordination with developers is an important aspect of a tester's daily activities.

Once the software tester identifies any bugs or discrepancies in the software, it is his/ her responsibility to communicate them to the developer and get them fixed.

### **Communication with different departments**

While coordination with developers is one aspect, what a test analyst does is also effective communication with other departments.

The software may not only have bugs in the development but also in graphics or other aspects.

## **8).Explain SDLC?**

The Software Development Life Cycle (SDLC) is a structured process that enables the production of high-quality, low-cost software, in the shortest possible production time.

The goal of the SDLC is to produce superior software that meets and exceeds all customer expectations and demands.

The SDLC defines and outlines a detailed plan with stages, or phases, that each encompass their own process and deliverables.

Adherence to the SDLC enhances development speed and minimizes project risks and costs associated with alternative methods of production.

## 9). Explain STLC?

STLC stands for Software Testing Life Cycle. STLC is a sequence of different activities performed by the testing team to ensure the quality of the software or the product.

STLC is an integral part of Software Development Life Cycle (SDLC). But, STLC deals only with the testing phases.

STLC starts as soon as requirements are defined or SRD (Software Requirement Document) is shared by stakeholders.

STLC provides a step-by-step process to ensure quality software.

In the early stage of STLC, while the software or the product is developing, the tester can analyze and define the scope of testing, entry and exit criteria and also the Test Cases.

It helps to reduce the test cycle time along with better quality.

As soon as the development phase is over, the testers are ready with test cases and start with execution.

This helps to find bugs in the initial phase.

## 10). What is a review and what is the objective of review?

A review is a systematic examination of a document by one or more people with the main aim of finding and removing errors early in the software development life cycle.

Reviews are used to verify documents such as requirements, system designs, code, test plans and test cases.

### **OBJECTIVE OF REVIEW:**

To improve the quality of the software product or component by identifying and resolving defects early in the development process, preventing issues from reaching end-users, leading to enhanced customer satisfaction and reduced support requests.

## 11). What are the different types of reviews?

Different types of Software Peer Reviews are there which are enumerated below:

**Code Review:** The source code of the software is examined here. The software is checked for the bugs and the bugs are removed from the code.

**Pair programming**: It is also a type of code review involving two people. The two people develop a code together at the same workstation.

**Inspection**: It is a formal type of review where a person has to go through a defined set of instructions in order to find defect/defects. There can be a number of reviewers involved in this type of reviewing.

**Walkthrough**: This is a process where the authors of the software as well as other associates are gathered at one place and they discuss about the software defects. Questions are made, comments are given, answers are given to all the queries people have regarding the software. With all the members' satisfaction, conclusions are made.

## **12). What is white Box Testing?**

**White Box Testing** is a testing technique in which software's internal structure, design, and coding are tested to verify input-output flow and improve design, usability, and security. In white box testing, code is visible to testers, so it is also called Clear box testing, Open box testing, Transparent box testing, Code-based testing, and Glass box testing.

## **13). What is Unit Testing?**

Unit Testing is a type of software testing where **unit Testing?** individual units or components of a software are tested. The purpose is to validate that each unit of the software code performs as expected. Unit Testing is done during the development (coding phase) of an application by the developers. Unit Tests isolate a section of code and verify its correctness. A unit may be an individual function, method, procedure, module, or object.

## **14).What is integration testing?**

Integration testing is the second level of the software testing process comes after unit testing. In this testing, units or individual components of the software are tested in a group. The focus of the integration testing level is to expose defects at the time of interaction between integrated components or units.

### **15).What is Black box Testing?**

Black box testing involves testing a system with no prior knowledge of its internal workings. A tester provides an input, and observes the output generated by the system under test. This makes it possible to identify how the system responds to expected and unexpected user actions, its response time, usability issues and reliability issues.

### **16).What is System Testing?**

*System Testing means testing the system as a whole. All the modules/components are Integrated in order to verify if the system works as expected or not.*

System Testing is done after Integration Testing. This plays an important role in delivering a high-quality product.

### **17).What is smoke testing and sanity Testing?**

Smoke testing is a crucial practice in software testing that allows you to quickly evaluate if the most important functions of your software are working correctly.

By conducting smoke tests, you can promptly identify mission-critical failures and address them early in the development process before diving into more intricate details.

This proactive approach enables you to ensure the smooth functioning of your software and enhance its overall quality.

### **18).What is the difference between smoke testing and sanity testing?**

Smoke Testing has a goal to verify “stability” whereas Sanity Testing has a goal to verify “rationality”.

Smoke Testing is done by both developers or testers whereas Sanity Testing is done by testers.

Smoke Testing verifies the critical functionalities of the system whereas Sanity Testing verifies the new functionality like bug fixes.

Smoke testing is a subset of acceptance testing whereas Sanity testing is a subset of Regression Testing.

Smoke testing is documented or scripted whereas Sanity testing isn't.

Smoke testing verifies the entire system from end to end whereas Sanity Testing verifies only a particular component.

### 19).What is retesting and regression testing?

#### RETESTING:

This testing is a part of a defect life cycle. In a defect life cycle, a bug cannot be closed until it has been solved by the developer and retested by the tester.

Once the tester retests **the** bug and confirms that it is solved, then only the bug can be closed. If the testers find that the bug is not solved then they will reassign it to the developer.

While retesting, a tester will recreate the same scenario to see whether the bug has been solved. A tester is also required to check other functionalities related to the bug to make sure new code changes for this bug are not altering other closely related functionalities.

#### REGRESSION TESTING:

Regression testing is a black box testing techniques. It is used to authenticate a code change in the software does not impact the existing functionality of the product. Regression testing is making sure that the product works fine with new functionality, bug fixes, or any change in the existing feature.

### 20).What is the difference between retesting and regression testing?

Regression Testing	Retesting
Involves testing a <i>general area</i> of the software.	Involves testing a <i>specific feature</i> of the software.
Is about testing software which <i>was</i> working, but now, due to updates, <i>might not</i> be working.	Is about testing software which you <i>know was not</i> working, but which you believe to have been fixed. You test it to confirm that it is now in fact fixed.
<i>Is ideal</i> for automation as the testing suite will grow with time as the software evolves.	<i>Is not ideal</i> for automation as the case for testing changes each time.
Should <i>always</i> be a part of the testing process and performed each time code is changed and a software update is about to be released.	Is <i>only</i> a part of the testing process if a defect or bug is found in the code.

## 21) What is end to end testing?

For those who are still exploring the world of Software Testing, E2E testing is when you validate your entire application from start to finish, along with any of its dependencies.

In E2E testing, you create an environment identical to the one that will be used by real users. Then you test all actions that your users might perform on your application.

With End-To-End testing, you test entire flows – like logging onto a website or buying a product from an online store.

End To End Testing **verifies complete system flow and increases confidence by detecting issues and increasing Test Coverage of subsystems**

## 22) What is system integration testing?

System **Integration testing** is defined as a type of software testing carried out in an integrated hardware and software environment to verify the behavior of the complete system. It is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirement.

## 23) What is vulnerability Testing?

Vulnerability testing refers to the process of evaluating your systems, software, and networks for potential weaknesses that could be exploited by cybercriminals. It helps you discover the flaws in your system before they can be exploited by malicious actors to gain unauthorized access to your business.

## 24) what is Efficiency Testing?

Efficiency testing is a type of performance testing that measures how well a software system can perform under different workloads and conditions. This testing is used to evaluate how efficiently a system can handle user requests, process data, and complete tasks. The goal of this testing is to identify any performance issues that may impact the system's ability to deliver results within an acceptable time frame.



## **25) what is entry criteria for system testing?**

**Some of the entry criteria, which are generally used to mark the beginning of the testing, are:**

- Complete or partially testable code is available.
- Requirements are defined and approved.
- Availability of sufficient and desired test data.
- Test cases are developed and ready.
- Test environment has been set-up and all other necessary resources such as tools and devices are available.

## **26) what is exit criteria for system testing?**

Some of the exit criteria:

- All functional and non-functional requirements must be met.
- All performance and load testing must be completed successfully.
- All security and reliability testing must be completed successfully.
- All critical test cases must be passed.
- Complete functional coverage must be achieved.
- All high-priority defects must be identified and fixed.
- All show stopper defects or blockers must be fixed and none of the identified critical/severity 1 defects should be in open status.

## **27) what is a test scenario?**

The test scenario is a detailed document of test cases that cover end to end functionality of a software application in liner statements. The liner statement is considered as a scenario. The test scenario is a high-level classification of testable requirements. These requirements are grouped on the basis of the functionality of a module and obtained from the use cases.

In the test scenario, there is a detailed testing process due to many associated test cases. Before performing the test scenario, the tester has to consider the test cases for each scenario.

In the test scenario, testers need to put themselves in the place of the user because they test the software application under the user's point of view. Preparation of scenarios is the most critical part, and it is necessary to seek advice or help from customers, stakeholders or developers to prepare the scenario.

## **28) what is a testcase?**

The test case is defined as a group of conditions under which a tester determines whether a software application is working as per the customer's requirements or not. Test case

designing includes preconditions, case name, input conditions, and expected result. A test case is a first level action and derived from test scenarios.

### **29) Explain ECP/BVA Techniques?**

Boundary Value Analysis (BVA) is a Black-Box testing technique used to check the errors at the boundaries of an input domain.

The name comes from the Boundary, which means the limits of an area. So, BVA mainly focuses on testing both valid and invalid input parameters for a given range of a software component.

If (Min,MAX) is the range given for a field validation, then the boundary values come as follows:

Invalid Boundary Check { Min-1 ; Max+1 }

Valid Boundary Check {Min; Min+1 ;Max-1;Max }

### **30) What is RTM?**

Traceability matrix is a table type document that is used in the development of software application to trace requirements. It can be used for both forward (from Requirements to Design or Coding) and backward (from Coding to Requirements) tracing. It is also known as **Requirement Traceability Matrix (RTM) or Cross Reference Matrix (CRM)**

### **31) Explain Defect/Bug Life cycle?**

**Defect Life Cycle** or Bug Life Cycle in software testing is the specific set of states that defect or bug goes through in its entire life. The purpose of Defect life cycle is to easily coordinate and communicate current status of defect which changes to various assignees and make the defect fixing process systematic and efficient.

### **32) Explain Defect severity and defect priority?**

Priority by the English definition is used in the comparison of two things or conditions, where one has to be given more importance than the other(s) and has to be tackled with/resolved first before proceeding to the next one(s). Therefore in the context of defects, the priority of a defect would indicate the urgency with which it would need to be fixed.

Severity by the English definition is used to describe the gravity of an undesirable occurrence. Hence when it comes to bugs, the severity of a bug would indicate the effect it has on the system in terms of its impact.

### **33) An example for a Defect which takes low severity but high priority?**

E.G: Improper logo or a spelling mistake in a company name

### **34) An example for a Defect which takes High severity but Low priority?**

**E.G:** A major issue belongs to module 2 and minor issue belongs to module 1.

### **35) Explain Agile Methodology?**

Agile Tutorial some times called agile methodology. This tutorial contains basic and advanced concepts of agile. Our Agile tutorial is designed for beginners and professionals. Agile is an iterative approach of software development methodology using short iterations of 1 to 4 weeks. Using Agile methodology, the software is distributed with fastest and fewer changes. The advantages of agile methodology are customer satisfaction by rapid, continuous development and delivery of useful software.

### **36) what is product Backlog?**

A product backlog is a list of the new features, changes to existing features, bug fixes, infrastructure changes, or other activities that a team may deliver in order to achieve a specific outcome.

The product backlog is the single authoritative source for things that a team works on. That means that nothing gets done that isn't on the product backlog. Conversely, the presence of a product backlog item on a product backlog does not guarantee that it will be delivered. It represents an option the team has for delivering a specific outcome rather than a commitment.

### **37) what is an Epic?**

In the scrum model, Epic defines a high-level expectation of a feature or software. For example, when you want to build software that keeps a track of your sales, an Epic can be "As a business owner, I want to build a software which I can use to keep the record of my sales". Now is the great Epic which tells about the purpose of software that needs to be built. More Epics can be derived from this great Epic which can further tell about the different modules of the application. Let's take another example that is derived from this great Epic and talk about the further modules of this software application.

Example: As a business owner, I want to see all the financial reports of my sales on a single dashboard.

### 38) what is user story?

A user story is a small, self-contained unit of development work designed to accomplish a specific goal within a product. A user story is usually written from the user's perspective and follows the format: "As [a user persona], I want [to perform this action] so that [I can accomplish this goal]."

### 39) what is sprint ?

A sprint is **a set amount of time that a development team has to complete a specific amount of work**. Sprints are generally planned to last about two weeks, though they can be as short as one week or as long as a month.

### 40) What is jira Tool?

Jira is a powerful work management tool designed for software teams that want to organize and track their projects.

It is an extremely flexible tool that can be customized to suit your team's unique workflows, meaning that teams of any kind can experience increased visibility and productivity when releasing new products or software.

As per Atlassian's data, over 65,000 companies worldwide have implemented Jira for its robust flexibility to support any project and its extensibility to work with various integrations and apps.

