1. What is testing?

Testing is a process within the software development life cycle which consists of verifying if an application or a program meets the specified requirements. The program is executed with the intent of finding any malfunction. Testing is essential in order to deliver a quality product.

1. Explain the testing principle “Test everything is not possible”

Exhaustive testing is impossible because we can’t test all the inputs or combinations of inputs to the program. Also, we can’t test all the paths that a user can follow. Instead, there are other testing techniques that assure good testing coverage (e.g.: using valid, invalid, null test data, equivalence class partitioning, boundary value analysis etc.)

1. Explain the testing principle “Early testing is important”

The sooner an inconsistency is found, the better. Time, effort and money can be saved by early testing. During the requirements analysis step, any ambiguity or inconsistency should be raised. The cost of bug fixing is higher in the later stage of development.

1. Explain the testing principle “Running many times the same tests/scenarios, will no longer find new defects”

The pesticide paradox says that if the same set of repetitive tests are conducted, no new defects will be discovered. That is because the bugs discovered at the first run of the test were fixed and the program is now immune to this set of tests. The solution is to review the test scenarios and add new test steps.

1. Explain the testing principle “Testing is done differently in different contexts”

Depending on the type of application, there are different methodologies, techniques and testing types to test a program. For instance, an e-commerce site will be tested differently than a banking application.

1. Explain the testing principle “ There is no bug-free application”

Even though the application was rigorous tested, there is no guaranty that there are no defects in the software. However, we can minimize the number and the severity of bugs using a large variety of testing types and techniques.

1. What is Component Testing?

Component testing involves testing each functionality separately by other components.

1. What is System Testing?

Testing of all components of a single system working integrated and according to the requirements. System end-to-end flows are covered.

1. What is Integration Testing?

Testing the interactions between the components of a system.

1. What is Acceptance Testing?

It is the final testing phase before deployment when real life scenarios are tested by the client from a business perspective based on acceptance criteria.

1. What is User Acceptance Testing UAT?

It is the final testing phase before deployment when real life scenarios are tested by the client from a business perspective based on acceptance criteria.

1. What is Alpha Testing?

Alpha testing is performed by the tester in a testing environment, before releasing the product.

1. What is Beta Testing?

Beta testing is performed by the user (client). The client gives a feedback about the product

1. What is Functional Testing?

Functional testing is based on functional requirements (what the system does).

Types of functional testing: smoke, integration, regression etc

1. What is Non-Functional Testing?

Non-functional testing is based on non-functional requirements (how the system does in terms of quality characteristics like performance, data loss, security, recovery)

Types of non-functional testing: volume, stress, performance, compatibility, security

1. What is Manual Testing?

The tests are run manually (without tools)

Advantages: does not require programming skills,

Downsides: time consuming, boring

1. What is Automated Testing?

The test are run using tools (Selenium, qtp, jmeter, soapui). Used on repetitive tests

Advantages: saves time, cost and effort

Downsides: require programming skills, hard to implement and maintain,

1. What is Black-box Testing? Examples

Test the functionality of the application without knowing its internal structure, code, based on requirements

Ex: smoke, compatibility, usability, end-to-end.

1. What is White-box Testing?

Testing of internal structures and code

Ex: code review, verification testing

Grey: Tests covering also the structure and the functionality of the system

1. What is Static Testing? Examples

Testing without running the application

Ex: code review, requirement analysis

1. What is Dynamic Testing? Examples

The software is compiled and executed

Ex: Exploratory, smoke, compatibility, regression etc

1. What is Usability Testing?

Verifies if the product is easy to use, intuitive.

1. What is Performance Testing?

Testing performed to determine how a system performs in terms of responsiveness, resource usage, reliability and stability under a particular workload. Subsets: load, stress, endurance, volume

1. What is Ad-hoc Testing?

Is a method of software testing without any planning and documentation. The tests are conducted informally and randomly without any formal expected results.

1. What is Smoke Testing?

is a type of software testing that comprises of a non-exhaustive set of tests that aim at ensuring that the most important functions work

1. What are Functional Requirements? Examples

* what the system does: create a login form with the following info: name, surname, phone, email, username, password

1. Give some Non-Functional Requirements examples

* how the system does: The page should load in maximum 400 msec; The system should allow 50k users simultaneous, compatibility

1. What is the requirements role?

Requirements describe what the software will do and how it is expected to perform. They must be clear, consistent, complete in order to understand what the client wants.

1. Why requirements analysis / clarifications?

Requirements describe what the software will do and how it is expected to perform. They must be clear, consistent, complete in order to understand what the client wants.

1. What is a Decision Table? For what is used?

Decision table testing is black box test design technique to determine the test scenarios for complex business logic when the number of input combination is very big.

1. What is a Test Environment?

Specific locations where the tests are run

1. What is a Cloud Test Environment? Examples of services

Infrastructure as a Service (IaaS): physical and virtual machines, servers, storage, load balancers, network

- Platform as a Service (PaaS): database, operating systems, development tools

- Software as a Service (SaaS): email, games, communication applications

1. Cloud Testing Advantages

Connect to the test environment anytime, it’s secure,

1. What is Test Data?

Specific data which is used in tests

1. What test data can be used for a good testing coverage?

Valid (Test the happy flows), invalid (Check the system behavior and that correct error messages are generated), null (Check that correct error messages are generated), volume data (Test the system with real data volume based on the application life cycle)

1. What are the advantages for production test data usage?

The test are more precise when using production test data. Also, performance testing leads to more accurate results.

1. What are the disadvantages for production test data usage?

Having test data in production will expose you to the risks (security holes)

1. What is and how is used the Equivalence Class Partitioning Technique?

A black-box technique used to divide the application inputs in sets (partitions) that can be considered the same, having the same output

Identify the system inputs

- Think what is the system output is for each input

- Eliminate inputs for which the system behaves the same

1. What is and how is used the Boundary Value Analysis Technique?

A black-box technique used to cover the partitions boundaries

- Identify the minimum and maximum values of each partition

- A test for each boundary is selected

1. What is an End User?

The end user is the person which uses the software after it has been developed.

1. What is a Happy flow?

The happy flow is the positive scenrario. The server errors, invalid data, connection issues are not taken into consideration.

1. What is a Test Case?

A set of steps and expected results developed for a particular test

1. Which are the Test Case Fields? Give details for each of it

Test Case ID – unique, used to identify easily a specific test case

• Test Case Name – summary of the test case

• Preconditions -

• Test Steps

• Expected results

• Status

• Defect

• Others: Priority, Req\_ID, Comments, Time

1. Test Case Management Tools Usage

Using test case management tools, the test cases are easy to create, execute and organize. The test cases are executed and the tools allows creating reports of failed and passed test cases

E: Testlink, Jira Zephir

1. Test Case Best Practices

Break down the app functionality in smaller measurable modules

Number of steps should be reasonable (Some of the steps can be written shortly and as one)

Keep it simple, easily to understand

Use a mix of positive and negative tests

1. What types of issues can you Boundary raise?
2. What is a Bug?

An error in an application or system which causes an incorrect result or unintended behavior • When the expected behavior and actual behavior are not matching

1. What is an improvement?

Adding new functionality or changing existing functionality in a way that makes the software application more efficient, faster, more usable, more useful, and/or more desirable

1. Bug Reporting Best Practices

Fields: description, precondition, steps to reproduce, actual results, expected results, screenshot, The environment on which the bug was identified, Reporter

To be short, clear, easy to understand. The feature can be added at the beginning

All details for actual and expected results

Report first the major bugs first

The bug description should reflect exactly the UI controls names and labels; e.g. dropdown list, Schedule Shut Down

1. What is Bug Severity?

The bug impact over the end-user – set by the tester –blocker, critical, major, normal, minor.

1. What is Bug Priority?

The importance and order a bug should be fixed. • Not set by tester • E.g. P1, P2, P3, P4 or low, normal, high, urgent, immediate

1. What is the difference between severity and priority?
2. What is Bug Affected Version?

The application version in which the defect was found.

1. What other information should you give while submitting a new bug?

Fields: description, precondition, steps to reproduce, actual results, expected results, screenshot, The environment on which the bug was identified, Reporter

1. What should you do before submitting a new bug?

Make sure the bug is not already reported

Make sure that the development changes are done

Make sure the issue can be reproduces

1. What is the Bug Description Template?

Fields: description, precondition, steps to reproduce, actual results, expected results, screenshot, The environment on which the bug was identified, Reporter

1. What is Exploratory Testing?

is an approach to software testing that is concisely described as simultaneous learning, test design and test execution. This testing is suitable if requirements and specifications are incomplete, or if there is lack of time

1. Do you know ways to measure the output of the exploratory testing?

Time boxed sessions with a declarative scope

Document the output of the session as you see fit

1. What are the advantages of doing exploratory testing?

Take less preparation

- Critical bugs are identified sooner

Simultaneous learning, test design, and test execution

Disadvantages?

- The tests cannot be reviewed - Difficult to keep track of what tests have been tested - Hard to repeat specific details of some earlier tests

1. What is Regression Testing?

Is a type of software testing that verifies that software previously developed and tested still performs correctly even after it was changed or interfaced with other software.

1. What are the advantages of doing regression testing?

Identify side effects due to bug fixes or other application changes

1. What Bug Validation?

Validate that the reported and fixed bugs are no more reproducible

1. What exactly you should test while doing bug validation?

Validate by priority & severity (if the priority is taken into account) or by severity

- Validate only the bugs for the current build and older

- The steps to reproduce are executed again on the fixed version, preferably on the same environment on which it was raised

- Validate the bug using various test data as inputs

- Test the bug related functionality to validate that nothing else was broken (no side effects)

- The bug is CLOSED if it does not reproduce anymore

- The bug is REOPEN if it is still reproducing or it is not fixed completely

- Provide valuable information related to the build version, test environment on which the bug validation was performed

1. What is Bug Resolution? Examples

The Resolution field indicates what happened to the bug.

FIXED

A fix for this bug is checked into the tree and tested.

INVALID

The problem described is not a bug.

WONTFIX

The problem described is a bug which will never be fixed.

DUPLICATE

The problem is a duplicate of an existing bug. When a bug is marked as a DUPLICATE, you will see which bug it is a duplicate of, next to the resolution.

WORKSFORME

All attempts at reproducing this bug were futile, and reading the code produces no clues as to why the described behavior would occur. If more information appears later, the bug can be reopened.

1. How would you prioritize your execution?

Run the BAT/Smoke tests first if existing

- Run the most important features first

Exploratory testing:

- Identify the major bugs first by testing the main flows first - Create a features importance list - Test top-down

Bug validation

Validate the higher severity first - Cluster and validate the related functionality bugs

1. What is a test management tool?

Using test case management tools, the test cases are easy to create, execute and organize. The test cases are executed and the tool allows creating reports of failed and passed test cases

E: Testlink, Jira Zephir

1. What test management tools do you know?

Testlink, Jira Zephir

1. What would you take into consideration while planning your execution?

Run the BAT/Smoke tests first if existing

- Run the most important features first

Exploratory testing:

- Identify the major bugs first by testing the main flows first - Create a features importance list - Test top-down

Bug validation

Validate the higher severity first - Cluster and validate the related functionality bugs

1. What is a Test Report?

The test report in a document which is delivered to the client at the end of a testing project. It contains information about the testing activities performed for the project, details about the quality of the product,

1. Test Report Types/Flavors
2. What a test report should contain?

**Purpose of the document (**Short description about the objective of preparing the document**)**

**Application Overview**

**Testing Scope (**This section explains about the functions/modules in scope & out of scope for testing; Any items which are not tested due to any constraints/dependencies/restrictions**)**

**Metrics (**test execution results, status of test cases & defects etc**)**

**Types of testing performed (**Describe the various types of Testing performed for the Project. This will make sure the application is being tested properly through testing types agreed as per Test Strategy**)**

[**Test Environment & Tools**](http://www.softwaretestinghelp.com/test-bed-test-environment-management-best-practices/) **(**Provide details on Test Environment in which the Testing is carried out. Server, Database, Application URL etc. If any Tools were used like Quality Center (now HP ALM) for logging defects**)**

**Lessons Learned (**This section is used to describe the critical issues faced and their solutions (how they were solved during the Testing). Lessons learnt will help to make proactive decisions during the next Testing engagement, by avoiding these mistakes or finding a suitable workaround**)**

**Recommendations**

**Best Practices**

**Exit Criteria**

**Conclusion/Sign Off**

1. What is SCRUM?

Scrum is a framework which helps organizing complex projects. Scrum has short fixed schedule of release cycles known as sprints. There are three chief roles in Scrum Testing – Product Owner, Scrum Master and The Development Team.

1. What are SCRUM documents, roles, process?
2. What is a Product Backlog?
3. What is a Sprint ?

It is a set period of time to complete the user stories, decided by product owner and developer team, usually 2-4 weeks of time.

1. What is Sprint Backlog?
2. Which are the Scrum Roles?

Product Owner - defines features of the product, prioritizes the features

* Scrum Master - manages the team and look after the team's productivity, Invites to the daily scrum, sprint review and planning meetings
* The Development Team = 5 -9 members, organizes and schedule their work on their own

1. What is discussed during Sprint Planning meeting?

A sprint begins with the team importing stories from the release backlog into the sprint backlog; it is hosted by scrum master. The Testers estimate effort to test the various stories in the Sprint Backlog.

1. What is discussed during Daily Scrum?

t is hosted by scrum master, it last about 15 minutes. During Daily Scrum, the members will discuss the work completed previous day, the planned work for the next day and issues faced during sprint. During daily stand-up meeting team progress is tracked.

1. What is discussed during Sprint Retrospective?

t is also hosted by scrum master, it last about 2-4 hours and discuss what team has accomplished in the last sprint and what lessons were learned.

1. What is a Release?

Activities that the development team can’t realistically complete within development sprints.

1. What is a Build?
2. Which is the last iOS version?

11

1. Which is the last Android version?

8.0

1. Do you know tools used testing on iOS?
2. Do you know tools used testing on Android?
3. What is localization testing?

ocalization Testing is a software testing technique, where the product is checked to assure that it behaves according to the **local culture or settings**

1. Give some mobile testing specifics?

Connectivity, notification, interruption, installation, location, 3rd party application integration testing

1. Testing types: ad-hoc testing vs exploratory testing

Ad-hoc Testing Exploratory Testing

Adhoc testing begins with learning application first and then work with actual testing process.

No formal test preparation takes place, no recognized test design technique is used, there are no expectations for results and arbitrariness guides the test execution activity

It works on negative testing mostly. This testing works on positive testing niche.

Exploratory Testing begins with the exploring the application while learning.

Documentation is mandatory in Exploratory Testing. To assure the quality it’s necessary to documents the detail of the testing.

This testing works on positive testing niche.

Regression Testing Retesting

Regression testing is a type of software testing that intends to ensure that changes like defect fixes or enhancements to the module or application have not affecting unchanged part

In Regression testing, you can include the test cases which passed earlier. We can say that check the functionality which was working earlier.

Regression test cases we use are derived from the functional specification, the user manuals, user tutorials, and defect reports in relation to corrected problems.

Retesting is done to make sure that the tests cases which failed in last execution are passing after the defects against those failures are fixed.

In Retesting, you can include the test cases which failed earlier. We can say that check the functionality which was failed in earlier build.

Test cases for Retesting cannot be prepared before start testing. In Retesting only re-execute the test cases failed in the prior execution.