Manual Testing

1. How does quality control differ from quality assurance?

• Quality control is a product-oriented approach of running a program to determine if it has any defects, as well as making sure that the software meets all of the requirements put forth by the stakeholders • Quality assurance is a process-oriented approach that focuses on making sure that the methods, techniques, and processes used to create quality deliverable are applied correctly.

2. What is Software Testing?

• Software Testing is a process used to identify the correctness, completeness, and quality of developed software. • It includes a series of activities conducted with the intent of finding errors in software so that it could be corrected before the product is released to the market.

3. Why is Software Testing Required?

Software testing is a mandatory process that guarantees that the software product is safe and good enough to be released to the market.

Here are some compelling reasons to prove testing is needed:

• It points out the defects and errors that were made during the development phases.

• Reduces the coding cycles by identifying issues at the initial stage of the development.

• Ensures that software application requires lower maintenance cost and results in more accurate, consistent and reliable results.

• Testing ensures that the customer finds the organization reliable and their satisfaction in the application is maintained.

• Makes sure that software is bug-free, and the quality of the product meets the market standard.

• Ensures that the application doesn’t result in any failures.

4. What are the two main categories of software testing?

Manual Testing

This is the oldest type of software testing where the testers manually execute test cases without using any test automation tools. It means the software application is tested manually by QA testers.

Automation Testing

This is the process of using the assistance of tools, scripts, and software to perform test cases by repeating pre-defined actions. Test Automation focuses on replacing manual human activity with systems or devices that enhance efficiency

5. What is alpha testing, beta testing, gamma testing, monkey testing and guerilla testing ?

Alpha Testing

It is a type of software testing performed to identify bugs before releasing the product to real users or to the public. Alpha Testing is a type of user acceptance testing.

Beta Testing

It is performed by real users of the software application in a real environment. Beta Testing is also a type of user acceptance testing.

Gamma Testing

It is the final stage of testing process conducted before software release. Focus on software security and functionality.

Monkey Testing

Is a type of software testing where software or application is tested using random inputs with the sole purpose of trying and breaking the system

Guerrilla Testing

Is performed on module based on some random inputs repeatedly and checks the modules functionality.

6. What are the different levels of manual testing?

Unit testing

It is a way of testing the smallest piece of code referred to as a unit that can be logically isolated in a system. It is mainly focused on the functional correctness of the standalone module.

Integration Testing

Individual software modules are integrated logically and tested as a group. Big-bang approach.

System Testing

In system testing all the components of the software are tested as a whole in order to ensure that the overall product meets the requirements specified. There are dozens of types of system testing, including usability testing, regression testing, and functional testing.

User Acceptance Testing

The final level, acceptance testing, or UAT (user acceptance testing), determines whether the software is ready to be released.

White box testing

Testing of software’s internal coding and infrastructure.

Black box testing

Just examines the functionality without bothering/peering about the code.

Retesting

Retesting is running the previously failed test cases again on the new software to verify whether the defects posted earlier are fixed or not.

7.What is Functional Testing?

Testing done to ensure that the product functions the way it is designed to according to the design specifications and documentation. Type of black box testing.

Different types in Functional Testing:

1. Smoke Testing

2. Sanity Testing

3. Regression Testing

Smoke Testing

This testing is performed on a system prior to being accepted for further testing. It is also called as Day0 testing. It is done when we receive a stable application with added feature.

Sanity Testing

Initial testing effort to determine if the core functionalities are performing well enough to accept it for further testing efforts.

Regression Testing

This is re-testing of the product/software to ensure that all reported bugs have been fixed and implementation of new changes has not affected the existing functionalities. These tests apply to all phases wherever changes are being made. This testing also ensures reported product defects have been corrected for each new release and that no new quality problems were introduced in the maintenance process

8. What's is non-functional testing?

Non-functional testing is focused on the performance of the entire in-scope solution. Every scenario executed against a system must fulfill the goals of the specific test and adhere to realistic user to business process ratios.

Types

1. Load Testing

2. Volume Testing

3. Stress Testing

4. Security Testing

5. Compatibility Testing

6. Migration Testing

7.Accessibility Testing

Load testing

Type of performance testing which determines the performance of a software, software product and system under real life-based load conditions. Basically, it deter mines the behavior of the application when multiple users use it at the same time.

Volume testing

Type of software testing, where the software is subjected to a huge volume of data to analyses the system performance.

Stress testing

Type of testing that verifies stability & reliability of software application to analyse robustness and error handling capacities under extremely heavy load conditions and ensuring that software doesn't crash under crunch situations.

Security Testing

It's a quality control activity to identify security defects or vulnerabilities in the software and verify if the software product has met its security requirements and its customer's security needs.

Compatibility Testing

To check the compatibility on different platform/environments.

Migration Testing

Is a verification process of migration of the legacy system to the new system with minimal disruption/downtime

Accessibility Testing

It's done to check the software is usable to as many people as possible. To accessible to those with disabilities, such as vision impairment, hearing disabilities, and other physical or other conditions.

9. What is a test bed in manual testing?

The test bed is an environment configured for testing. It is an environment used for testing an application, including the hardware as well as any software needed to run the program to be tested. It consists of hardware, software, network configuration, an application under test, other related software.

10. Explain the procedure for manual testing?

1. Planning and Control

2. Analysis and Design

3. Implementation and Execution

4. Evaluating exit criteria and Reporting

5. Test Closure activities

11. What is the test case?

Test case is a document that has a set of conditions or actions that are performed on the software application to verify the expected functionality of the feature. Test cases describe a specific idea that is to be tested, without detailing the exact steps to be taken or data to be used. For example, in a test case, you document something like 'Test if coupons can be applied on actual price'.

12. What's the difference between verification and validation in testing?

Verification It is a static analysis technique. Here, testing is done without executing the code. Examples include – Reviews, Inspection, and walkthrough. Validation It is a dynamic analysis technique where testing is done by executing the code. Examples include functional and non-functional testing techniques.

13. What's the difference between a bug and a defect?

A bug is a just fault in the software that's detected during testing time. A defect is a variance between expected results and actual results, detected by the developer after the product goes live.

14. What are the advantages of manual testing?

- It is a cheaper way of testing when compared to automated testing
- Analysis of product from the point of view of the end-user is possible only with manual testing
- GUI testing can be done more accurately with the help of manual testing as visual accessibility and preferences are difficult to automate
- East to learn for new people who have just entered testing
- It is highly suitable for short-term projects when test-scripts are not going to be repeated and reused for thousands of times
- Best suited when the project is at the early stages of its development
- Highly reliable, since automated tests can contain errors and missed bugs

15. Bugs Lifecycle

New: When a new defect is logged and posted for the first time. It is assigned a status as NEW.

Assigned: Once the bug is posted by the tester, the lead of the tester approves the bug and assigns the bug to the developer team

Open: The developer starts analyzing and works on the defect fix

Fixed: When a developer makes a necessary code change and verifies the change, he or she can make bug status as "Fixed".

Pending retest:

Once the defect is fixed the developer gives a code for retesting the code to the tester. Since the software testing remains pending from the testers end, the status assigned is “pending retest.”

Retest:

Tester does the retesting of the code at this stage to check whether the defect is fixed by the developer or not and changes the status to “Re-test.”

Verified:

The tester re-tests the bug after it got fixed by the developer. If there is no bug detected in the software, then the bug is fixed, and the status assigned is “verified.”

Reopen:

If the bug persists even after the developer has fixed the bug, the tester changes the status to “reopened”. Once again, the bug goes through the life cycle.

Closed:

If the bug is no longer exists then tester assigns the status “Closed.”

Duplicate:

If the defect is repeated twice or the defect corresponds to the same concept of the bug, the status is changed to “duplicate.”

Rejected:

If the developer feels the defect is not a genuine defect then it changes the defect to “rejected.”

Deferred:

If the present bug is not of a prime priority and if it is expected to get fixed in the next release, then status “Deferred” is assigned to such bugs

Not a bug:

If it does not affect the functionality of the application then the status assigned to a bug is “Not a bug”.

16. What is API testing?

API testing is a type of software testing where application programming interfaces (APIs) are tested to determine if they meet expectations for functionality, reliability, performance, and security. In simple terms, API testing is intended to reveal bugs, inconsistencies or deviations from the expected behavior of an API.

API Testing have three separate layers:

• Presentation Layer or user interface

• Business Layer or application user interface for business logic processing

• Database Layer for modeling and manipulating data

• API testing is performed at the most critical layer of software architecture, the Business Layer.

17. Rest Assured API.

REST Assured is a Java library for testing RESTful APIs. It is widely used to test JSON and XML based web applications. Furthermore, it fully supports all methods including the GET, PUT, POST, PATCH, and DELETE. Still, you are required to have Java, Maven, TestNG, and IDE (IntelliJ, Eclipse, etc.)

18. Types of Requests

Method	Description
GET	Fetch status line, Response body, Header etc.
HEAD	Same as GET, but only fetch status line and header section
POST	Perform request using request payload mostly in creating a record at the sever
PUT	Useful in manipulating/updating the resource using Request payload
DELETE	Deletes information relating to the target resource.
OPTIONS	Describe the communication options for the target resource
PATCH	Very much similar to put but it is more like a minor manipulation of resource content

19. Response Code

1. 100 Series

These are temporary Responses

100 Continue

101 Switching Protocols

102 Processing

2. 200 Series

The client accepts the Request, being processed successfully at the server.

200 – OK

201 – Created

202 – Accepted

203 – Non-Authoritative Information

204 – No Content

205 – Reset Content

206 – Partial Content

207 – Multi-Status

208 – Already Reported

226 – IM Used

3. 300 Series

Most of the codes related to this series are for URL Redirection.

300 – Multiple Choices

301 – Moved Permanently

302 – Found

303 – Check Other

304 – Not Modified

305 – Use Proxy

306 – Switch Proxy

307 – Temporary Redirect

308 – Permanent Redirect

4. 400 Series

These are specific to client-side error.

400 – Bad Request

401 – Unauthorised

402 – Payment Required

403 – Forbidden

404 – Not Found

405 – Method Not Allowed

406 – Not Acceptable

407 – Proxy Authentication Required

408 – Request Timeout

409 – Conflict

410 – Gone

411 – Length Required

412 – Precondition Failed

413 – Payload Too Large

414 – URI Too Long

415 – Unsupported Media Type

416 – Range Not Satisfiable

417 – Expectation Failed

418 – I’m a teapot

421 – Misdirected Request

422 – Unprocessable Entity

423 – Locked

424 – Failed Dependency

426 – Upgrade Required

428 – Precondition Required

429 – Too Many Requests

431 – Request Header Fields Too Large

451 – Unavailable For Legal Reasons

5. 500Series

These are specific to the server-side error.

500 – Internal Server Error

501 – Not Implemented

502 – Bad Gateway

503 – Service Unavailable

504 – Gateway Timeout

505 – HTTP Version Not Supported

506 – Variant Also Negotiates

507 – Insufficient Storage

508 – Loop Detected

510 – Not Extended

511 – Network Authentication Required

20. Best practices for testing

• Adopt a Controlled Security Test Environment with a Dedicated Team

• Understand Product Objectives in Order to Design Effective Test Strategies

• Proactively Plan Software Test Cycles

• Use Development Practices That Are Test-Oriented

• Ensure All Tests Are Integrated in CI/CD Pipeline

• Adopt Tests Written for Maximum Coverage

• Run Regular QA Technical Reviews

• Enact User Acceptance Testing (UAT)

• Measure Code Quality.

• Develop Requirements That Are Testable

21. Stages of SDLC.

1. Requirement gathering and analysis.

2. Define - Software Requirement Specification (SRS)

3. Design - Design Document Specification (DDS)

4. Implementation or coding or Built

5. Testing.

6. Deployment.

7. Maintenance

22. Stages of STLC

1. Requirement Analysis

2. Test Planning and Control

3. Test Analysis and Design

4. Test implementation and execution

5. Evaluating exit criteria and Reporting

23. What is Severity and Priority?

• Severity of a defect is impact of the bug terms of financial loss, damage to environment, company's reputation and loss of life.

• Priority of a defect is related to how quickly a bug should be fixed and deployed to live. • Severity is set by the Tester.

• Priority is set by the Customer.

24. Severity and Priority Levels.

High Severity –High Priority bug

This is when major path through the application is broken

Ex: Flipkart, every customer got error message when placing orders and the order is not taken.

Low Severity – High Priority bug

Functionally all looks good but due to minor non-functional part, the business reputation will be lost.

Ex: the logo or name of the company is wrongly displayed on the website. It is important to fix the issue as soon as possible, although it may not cause a lot of damage.

High Severity – Low Priority bug

This happens when the bug causes major problems, but it only happens in very rare conditions or situations

Ex: Customers who use Internet Explorer 8 cannot continue with their purchase of a product. Because the number of customers with very old browsers is very low, it is not a high priority to fix the issue.

Low Severity – Low Priority bug

When the bug doesn’t cause disaster (Showstopper) and only affects very small number of customers, both Severity and Priority are assigned low

Ex: Cosmetic defects, design issues and text overlapping