

**1. Discuss the significance of understanding different types of testing methodologies in ensuring software quality.**

Comprehending various testing approaches is essential for guaranteeing software quality since it aids in early defect detection, requirement compliance, and performance, security, and reliability improvement. Every methodology has a distinct function, such as confirming user satisfaction (UAT), examining system-wide functionality (system testing), or validating individual components (unit testing). An organized testing strategy lowers the risks and expenses related to production flaws.

**2. Compare and contrast manual testing and automated testing.**

In manual testing, test cases are carried out by testers by hand without the aid of automation tools. It works best for small projects with changing test cases, exploratory testing, and usability testing. Although it permits flexibility and adaptability because human testers conduct the testing, it can be time-consuming and subject to human error.

Automated testing, on the other hand, uses tools and scripts to run test cases automatically. Repetitive tasks that require efficient execution are best suited for large projects, regression testing, and performance testing. Compared to manual testing, automated testing is quicker, more accurate, and more scalable; however, it necessitates a larger initial investment in test automation tools and script development. Although automation increases productivity, manual labor cannot be entirely replaced by it.

**3. Explain when it is appropriate to use manual testing versus automated testing, and vice versa.**

Use Manual Testing When:

Conducting ad hoc, usability, and exploratory testing

Testing one-time tests or small-scale projects

Verifying visual elements (such as UI coherence)

Human judgment and experience are necessary for testing.

#### Use Automated Testing When:

Performing iterative regression tests  
Performing stress, load, and performance tests  
Testing on various devices or environments  
Cutting down on testing time for extensive projects

#### **4. Define functional testing and discuss its purpose in software testing.**

Functional testing confirms that software operates in accordance with its stated specifications. It guarantees that functions like data processing, form submission, and user authentication operate as intended. Ensuring accuracy, completeness, and adherence to business logic is the goal of functional testing.

#### **5. Describe different techniques used in functional testing**

Unit testing: Examines each software component separately.  
Integration testing: Confirms how integrated components interact with one another.  
System testing verifies that the entire system complies with the specifications.  
Prior to more thorough testing, smoke testing verifies fundamental functionality.  
Sanity testing: Verifies that no significant flaws have been introduced by recent changes.  
User Acceptance Testing (UAT): Verifies that the program satisfies user requirements.

#### **6. Define non-functional testing and explain its importance in assessing the quality attributes of software.**

Instead of focusing on particular functionalities, non-functional testing assesses software's quality attributes like performance, security, and usability. It guarantees the software's usability, security, and scalability. Meeting performance goals, avoiding security flaws, and guaranteeing a seamless user experience all depend on this testing.

#### **7. Discuss common types of non-functional testing, including performance testing, security testing, and usability testing.**

System speed, responsiveness, and stability under load are all measured by performance testing.  
Security testing guarantees data protection and finds vulnerabilities.

Usability testing evaluates how user-friendly and intuitive the program is.

Reliability testing (ensuring consistent performance over time) and compatibility testing (examining system behavior across devices/browsers) are additional types.

### **8.Explain the concept of regression testing and its role in software maintenance**

Regression testing makes sure that updates to the code don't cause problems with already-existing features. It is carried out to ensure that earlier features continue to function properly following bug fixes, improvements, or integrations. Stability and ongoing software maintenance depend on this kind of testing.

### **9.Define integration testing and discuss its significance in verifying interactions between software components.**

Integration testing examines how integrated software modules interact with one another to ensure proper operation. It guarantees that data moves between external systems, APIs, and components in an appropriate manner. This testing is essential for identifying problems like inconsistent data, poor communication, and erroneous module dependencies.

### **10.Define system testing and discuss its purpose in evaluating the entire software system.**

System testing verifies the overall functionality of the software system. It guarantees that every element, both functional and non-functional, functions as intended. Prior to deployment, this kind of testing aids in identifying system-wide flaws.

### **11.Define user acceptance testing (UAT) and explain its role in validating software from an end-user perspective.**

The last stage of testing is called User Acceptance Testing (UAT), during which actual users verify that the program meets their business requirements. Before the software is released, UAT assists in locating usability problems, missing requirements, and inconsistencies with real-world workflows. The software is prepared for production if it passes UAT.

### **12.Compare and contrast smoke testing and sanity testing. 13.Explain when each type of testing should be conducted during the software development lifecycle.**

Although they are both initial testing techniques used to guarantee software stability, smoke testing and sanity testing have distinct functions. A comprehensive, high-level test called "smoke testing" is carried out following a new software build to confirm that the essential features are operational. It functions similarly to a basic system health check, making sure that there are no significant problems before moving on to more thorough testing. Additional testing is stopped until the significant flaws are fixed if smoke testing is unsuccessful.

Sanity testing, on the other hand, is a targeted examination carried out following bug fixes or small updates to make sure that particular features impacted by recent modifications are operating as intended. Since it only examines the altered areas and not the entire system, it is a more focused and efficient procedure than smoke testing. The development team must rework the impacted features and cease additional testing if sanity testing is unsuccessful.

### **13.Explain when each type of testing should be conducted during the software development lifecycle.**

To guarantee software quality and stability, different kinds of testing are carried out at different phases of the software development lifecycle (SDLC). During the development stage, the procedure usually begins with unit testing, in which separate software components are examined separately. Integration testing is done after unit testing is finished to ensure that various modules work together properly.

System testing is carried out to assess the entire software against requirements following integration testing. A smoke test is performed to verify the stability of the most recent build before moving on to system testing. In-depth system testing is carried out if the smoke test is successful. Sanity testing is done whenever small adjustments or bug fixes are made to make sure they haven't resulted in unforeseen problems.

To make sure that current functionalities are not impacted by code changes, regression testing is carried out. Prior to deployment, performance and security testing is typically carried out to evaluate non-functional elements like speed, scalability, and security flaws. Lastly, before the software is made available to end users, user acceptance testing (UAT) is carried out to make sure it satisfies their needs and expectations.