**ID: 27377**

**NAME: Vinay Kumar G**

**Q. Brief Summary on Any Five Different Testing Styles**

1. **Black Box Testing:**

* Black Box Testing is a technique where the tester examines the functionality of an application without any knowledge of its internal code, structure, or implementation details.
* The main objective of this testing is to check whether the software meets the specified requirements by providing different inputs and observing the outputs.
* Testers do not concern themselves with how the application processes the inputs internally.
* Instead, they verify whether the results match the expected outcomes.
* This type of testing is typically used in system testing, acceptance testing, and functional testing phases.

Key points:

* No knowledge of internal code is required.
* Focuses on functionality.
* Helps uncover missing functionalities, incorrect outputs, and interface issues

**Example:**  
Testing a login page: Entering the username and password to check if login succeeds without looking at backend code.

**Popular Tools:**

* Selenium
* QTP (UFT)

1. **White Box Testing:**

* White Box Testing, also known as clear box or structural testing, requires the tester to have a deep understanding of the internal workings of the application.
* Test cases are created based on the code structure, logic, and data flow within the system.
* The main aim is to verify the internal operations of the system, ensuring all paths, branches, loops, and conditions in the code are properly tested.
* This style is mostly applied during unit testing, where developers themselves usually perform the tests.
* It is useful for identifying hidden errors in the code, logical errors, and security vulnerabilities.
* White Box Testing improves code optimization, error handling, and performance tuning, helping deliver a more robust and secure application.

Key points:

* Requires code knowledge.
* Tests all logical paths, branches, and loops.
* Helps detect hidden errors, improve code quality, and ensure security.

**Example:**  
Testing whether all "if-else" conditions in a payment calculation module are covered.

**Popular Tools:**

* JUnit (for Java)
* NUnit (for .NET)
* PyTest (for Python)

1. **Regression Testing:**

* Regression Testing is a type of testing that ensures recent code changes have not negatively affected the existing features of the software.
* When new functionalities are added, bugs are fixed, or enhancements are made, there is a risk that these changes could introduce new errors into the system.
* Regression tests are performed to verify that the unchanged parts of the application still behave as expected.
* This testing is crucial for maintaining the overall stability of the application after updates.
* It is often automated to save time, especially in large projects where manual regression testing would be very time-consuming.
* Regular regression testing helps ensure software reliability over time.

Key points:

* Maintains software stability after changes.
* Often automated for quick feedback.
* Essential after bug fixes, updates, and enhancements.

**Example:**  
After fixing a bug in the registration form, testing the full user registration flow again to confirm old features still work.

**Popular Tools:**

* Selenium
* JUnit with CI/CD
* TestNG

1. **Smoke Testing:**

* Smoke Testing is a preliminary testing process to check whether the basic and critical functionalities of an application are working correctly.
* It is often called "build verification testing," as it ensures that the build is stable enough for further, more detailed testing.
* Smoke Testing is done immediately after a new build is released. If the basic features work correctly, the build is accepted for further testing; if not, the build is rejected and sent back for fixes.
* This type of testing helps detect major issues early, saving time and effort by preventing detailed testing of unstable builds.
* It acts as a checkpoint before moving into deeper testing phases.

Key points:

* Basic and essential functionality check.
* Performed after receiving a new build.
* Saves time by detecting critical failures early.

**Example:**  
Checking if the website loads, the login page opens, and main buttons work after a fresh deployment.

**Popular Tools:**

* Jenkins (for automated smoke tests)
* Selenium
* Bamboo

1. **Exploratory Testing:**

* Exploratory Testing is a flexible and creative testing approach where testers actively learn and explore the software while testing it.
* Instead of following predefined test cases, testers use their skills and intuition to discover defects and understand the application's behaviour.
* This testing style is valuable when documentation is incomplete or when quick feedback is required.
* It often leads to the discovery of hidden bugs, usability issues, and unexpected behaviours that might not be found through scripted testing.

Key points:

* No scripted test cases.
* Tester learns and tests simultaneously.
* Good for uncovering hidden defects and usability issues.

**Example:**  
Exploring a new e-commerce app to find unexpected bugs by trying random product searches, cart actions, and payments.

**Popular Tools:**

* Testpad
* Xray (for exploratory sessions)
* Session-Based Test Management (SBTM) tools.

