Terminology

IT = Information Technology

Dev = Developer

PM = Project Manager

BA = Business Analyst

QA = Quality Assurance

TAE = Test Automation Engineer

STLC = Software Testing Life Cycle

Macro-Levels

Black Box Testing is performed without knowledge of internal operation. This does not only refer to getting to know the source code but rather implies not having seen any of the software documentation.

The objective is to get users to interact with the software in more manual way.

White Box Testing, also known as structural tests or tests based on internal logic of a program, is center in evaluating the source code internal of a software application.

In the white box testing examines the structure, logic, and paths of the code.

Characteristics

- Knowledge of source code.
- Approach in internal logic.
- Design of specific test cases.
- Covering code.
- Identification of errors in the code.

Grey Box Testing is a combination of black box testing and white box testing. The objective of this type of testing is to find defects due to incorrect structure or incorrect use of applications.
A tester of grey box testing partially knows the internal structure, which includes access to documentation on internal data structures and the algorithms used.

The testers of grey box testing needed documentation that describes application in both detail and at an important level, which brings together for test case define.

Software Development Life Cycle (SDLC)
Requirements analysis
Test planification's
Development of test cases
Configuration of test environment
Testing
End of the testing cycle

Requirements Analysis
This phase is where the testing team studies requirements from a testing perspective to identify testable requirements, and the quality assurance team can interact with various stakeholders to understand the requirements in detail. These requirements can be functional or non-functional. The feasibility of automation for the testing project is also performed at this design stage.

Requirements Phase Activities

- Identify the type of tests that are performed.
- Gather the details about priorities and approach.
- Prepare the requirements traceability matrix (RTM).
- Identify the details of the test environment where the test is supposed to be conducted.
- Analysis of feasibility of automation (If is necessary).

Planning Test

Set of agreements on how testers will conduct testing during a project.
These are normally recorded in an official document that provides insight into the flow of the distinct phases of the project, associated with software quality process.

Test Plan Phases

1. Objectives and scope/ Preparation of test planning.
2. Necessary resources.
3. Testing strategies.
4. Responsibilities and roles.
5. Entry and exit criteria.

The importance of testing plan lies in the fact that it provides clear and structured guidance for the testing process, ensuring that all relevant areas are covered and that software quality objectives are met.

It helps identify and allocate resources efficiently.

Allow best coordination and communication between members of testing team.

Facilitates the evaluation and continuous improvement of the testing process by providing a basis for reviewing and adjusting the testing strategy as necessary throughout the project.

Test Case Development

The test case development phase involves the creation, verification and reworking of test cases and test scripts once the test plan is ready. Initially, test data is identified, then created and reviewed and then reworked based on the preconditions. Next, the quality assurance team begins the process of developing test cases for individual units.

Test Case Development Activities

- Create test case, and/or automation scripts.
- Review and test cases / reference scripts.
- Create test data (if the environment is available).

Fundamental Principles of Testing (ISTQB)

1. **Testing shows presence of defects, not its absence**

The purpose of testing is to find errors. Even if a system passes all tests, it cannot be guaranteed to be free of defects.

2. **Exhaustive testing is not possible**

It is not feasible to test all possible combinations. It is necessary to apply techniques for selecting and prioritizing test cases.

3. **Early testing saves time and money**

Detecting errors early on (requirements, design) reduces costs and rework. This is the approach of Shift Left Testing.

4. **Grouping defects**

Errors tend to be concentrated in certain modules of the system. More effort should be focused on those areas.

5. **The pesticide paradox**

Repeating the same tests continuously reduces their effectiveness. Tests must be updated to continue detecting errors.

6. **Tests depend on context**

There is no single way to test. The approach depends on the type of product, risks, industry, and environment.

7. **The absence of errors is not guarantee of quality**
   An error-free system can still fail if it does not meet user needs or business requirements.

Testing Macro-Types

Functional testing: allow you to analyze business requirements.
Non-functional testing: analyze the approach of user experience.
Maintenance testing: are tests of changes to an operating system or the impact of a modified environment on an operating system.
UAT testing (User Acceptance testing): are an essential step in software development, because they are the final test before launch.
UI/UX Testing: are a series of actions used for the performance measure and general functionality of visual elements application. to verify and validate various user interface functions and ensure that there are no unexpected results, defects or errors.
Unit Testing: Those performed on individual software blocks can be automated and are normally carried out during the development phase.
Service/Integration Testing: These are made by covering multiple relational functionalities (both APIs and databases integrated).
Smoke Testing: These are tests that are run after a compilation phase to prevent failures or the application from "crashing". They are a type of initial and superficial test.

NOTE: Within the context of functional testing levels (UAT, UI Testing), work is carried out at the macro level of black box testing, while (integration testing, unit testing) is carried out at the macro level of white box testing.

Testing Functional

- Exploratory testing
- GUI testing (manual or automation)
- API testing (manual or automation)
- Database testing (manual or automation)

Strategic type

- End to End (E2E) (manual or automation): testing from frontend to backend.
- Re-testing: testing verify bugs.
- Regression testing: can include smoke testing and sanity testing.
- Smoke testing (for Unit testing): regression tests to validate a specific component and environment.
- Sanity testing (for Unit testing): regression tests to validate a specific component or area (subset of smoke test).

Testing non-functional

- Security testing
- Availability testing
- Efficiency testing
- Integrity testing
- Reliability testing
- Survivability testing
- Usability testing
- Performance testing
- Load/Stress testing

Most Common Types of Tests

Acceptance testing: Verify that the system meets customer requirements.

Usability testing: Evaluate the use ability and user experience to interact with system.

Load testing: Evaluate the system's performance under maximum load conditions, verifying its responsiveness and performance.

Security testing: Identify system vulnerabilities, ensuring data protection and integrity.

Stress testing: Evaluate the behavior of the system's performance under extreme stress conditions to determinate its resistance.

Location testing: are based on identifying illustrations (maps, diagrams, graphs, drawings or photographs) in which the student must locate or identify elements.