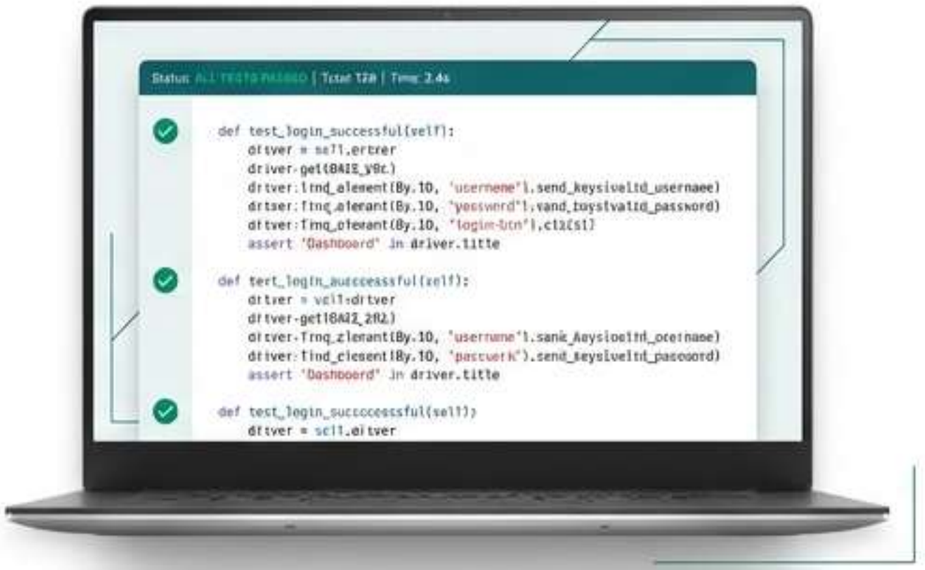
Testleaf
Always Ahead



Comprehensive Software Testing Strategies

Methodologies, Processes, and Best Practices for Quality Assurance

CORPORATE TRAINING SERIES

Session Agenda

- Introduction to Software Testing
- Core Concepts & Taxonomy
- Detailed Testing Methodologies
- The Testing Lifecycle & Process
- Real-World Scenarios
- Advanced Implementation Challenges
- Interview Preparation & Key Takeaways
- Knowledge Assessment

Testleaf
Always Ahead

The Role of Software Testing

Definition

The process of evaluating software to verify it functions as expected and meets specific requirements.

Primary Objective

Identify defects, ensure stability, and deliver a high-quality user experience.

Scope

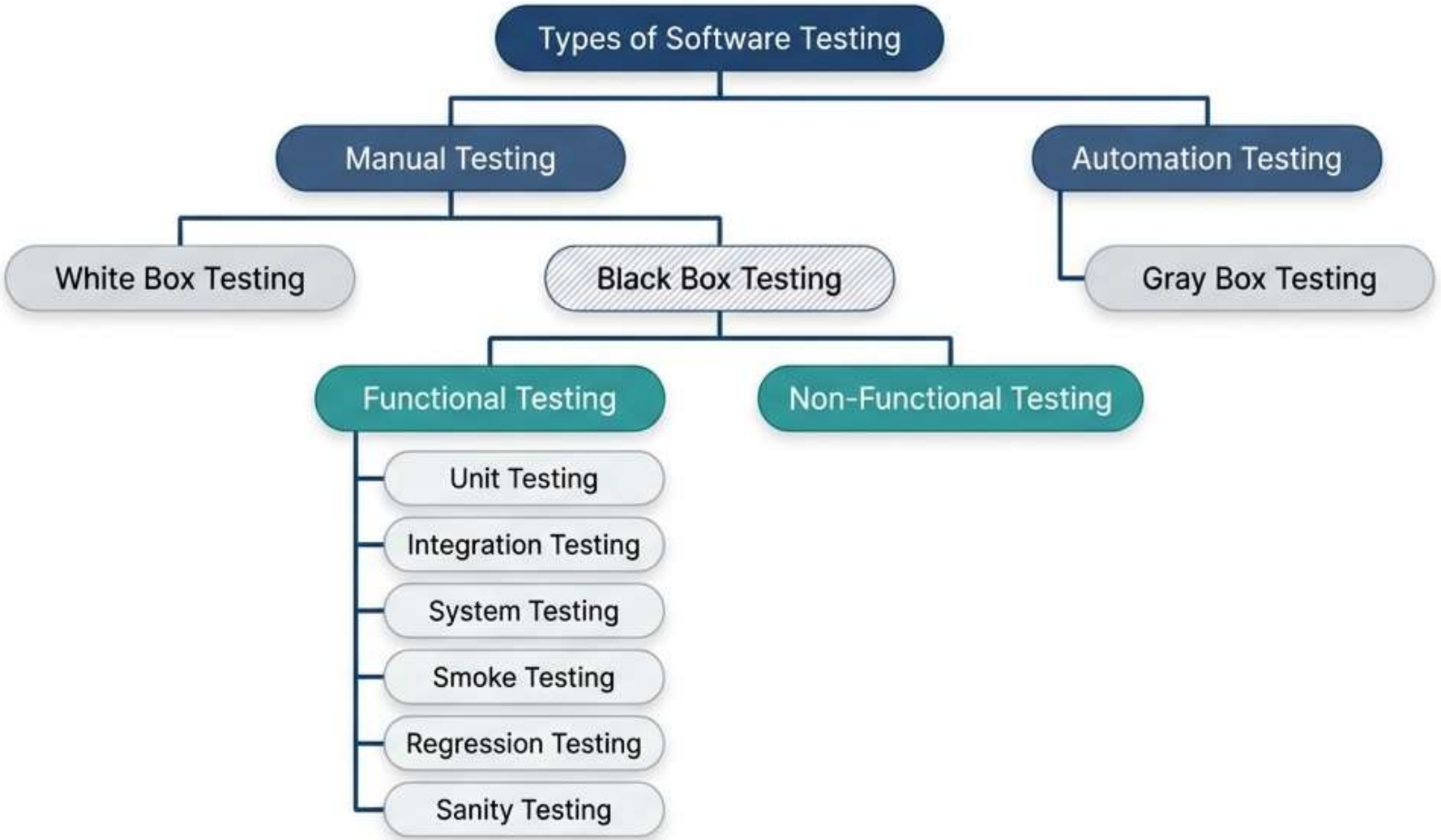
Encompasses verifying logic (code), behaviour (functionality), and performance (scalability).

Key Value

Reduces risk of failure in production and ensures business continuity.

Testleaf
Always Ahead

Testing Taxonomy & Core Classifications



Manual vs. Automation Testing

Manual Testing

Focus: Exploratory testing, usability, and ad-hoc scenarios.

Role: Tester simulates end-user behaviour to identify defects visually and functionally.

Automation Testing

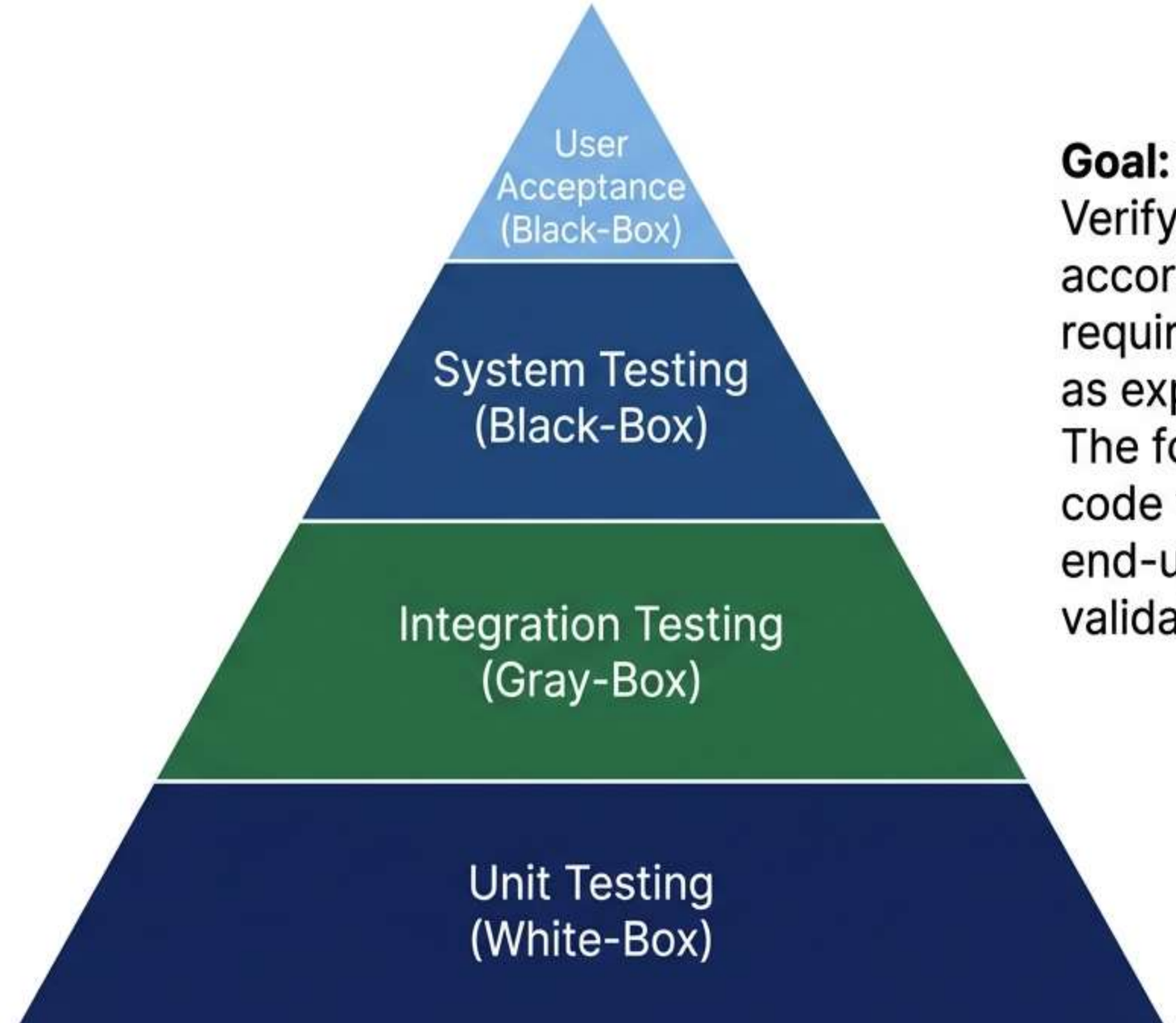
Focus: Regression, load testing, and repetitive tasks.

Mechanism: Uses scripts (Java, Python, C#) and tools to execute pre-defined test cases.

Benefit: Increases efficiency, consistency, and repeatability of the testing cycle.

Functional Testing Breakdown

Testleaf
Always Ahead



Goal:
Verify the system functions according to specified requirements and behaves as expected.
The foundation is built on code verification, rising to end-user workflow validation.

Testleaf
Always Ahead

The Testing Stability Workflow

Build Verification (Smoke)

Targeted Verification (Sanity)

Safety Net (Regression)

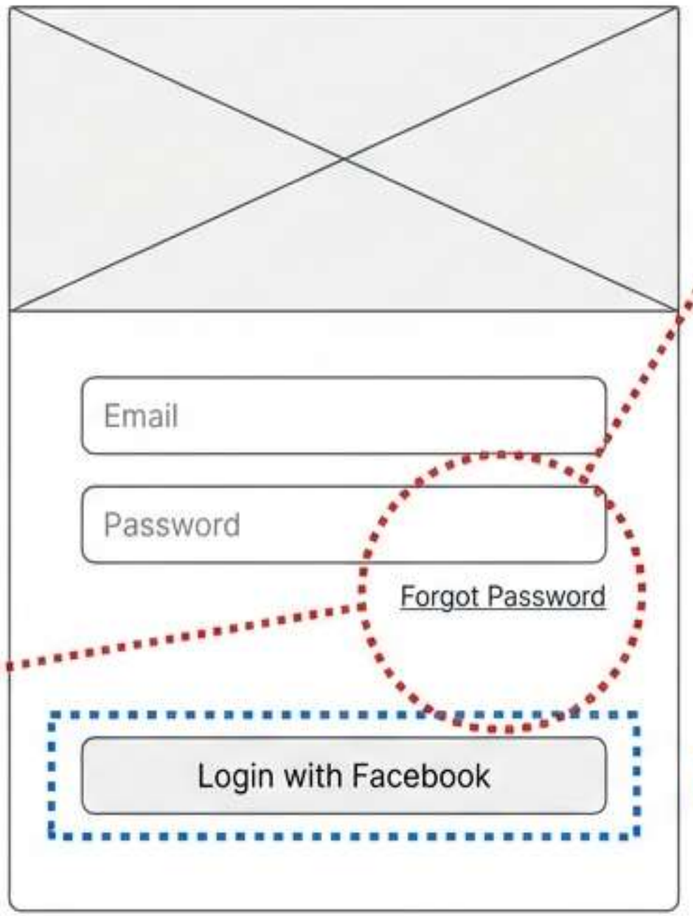
High-level check of critical functionalities. Determines if build is stable.

Performed after minor changes/fixes. Verifies specific components.

Comprehensive verification. Ensures updates didn't break existing features.

Applied Scenarios: Authentication Module

Scenario: A web application updates its login page.



Sanity Testing Scope (Fix Verification)

Sanity Testing Scope (Fix Verification)

Sanity verifies the specific fix.

Regression ensures the new Facebook button hasn't broken the standard Email login.

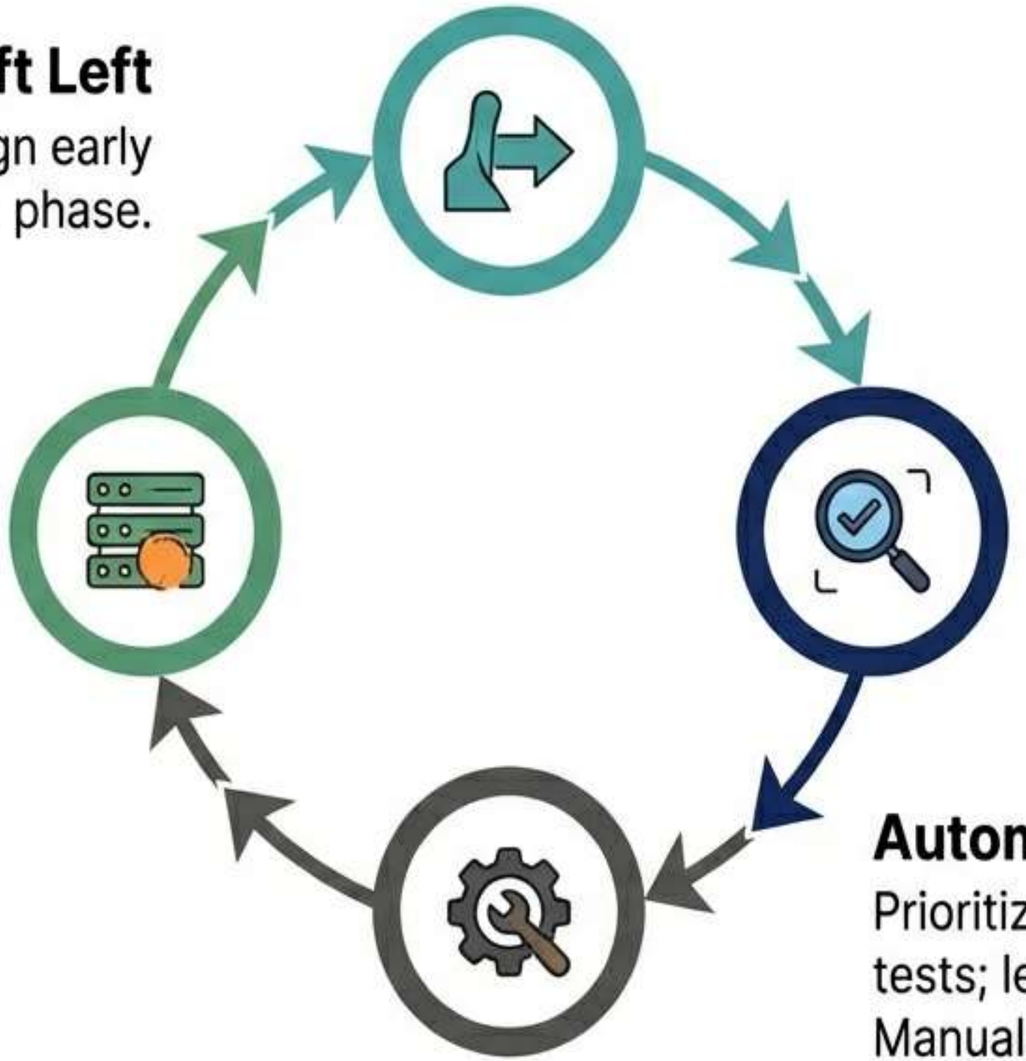
Regression Testing Scope (New Feature Impact)

Testleaf
Always Ahead

Corporate Testing Best Practices

Shift Left

Engage in testing design early in the requirement phase.



Define Scope

Clearly differentiate between Smoke (Build health) and Sanity (Feature health).

Automate Wisely

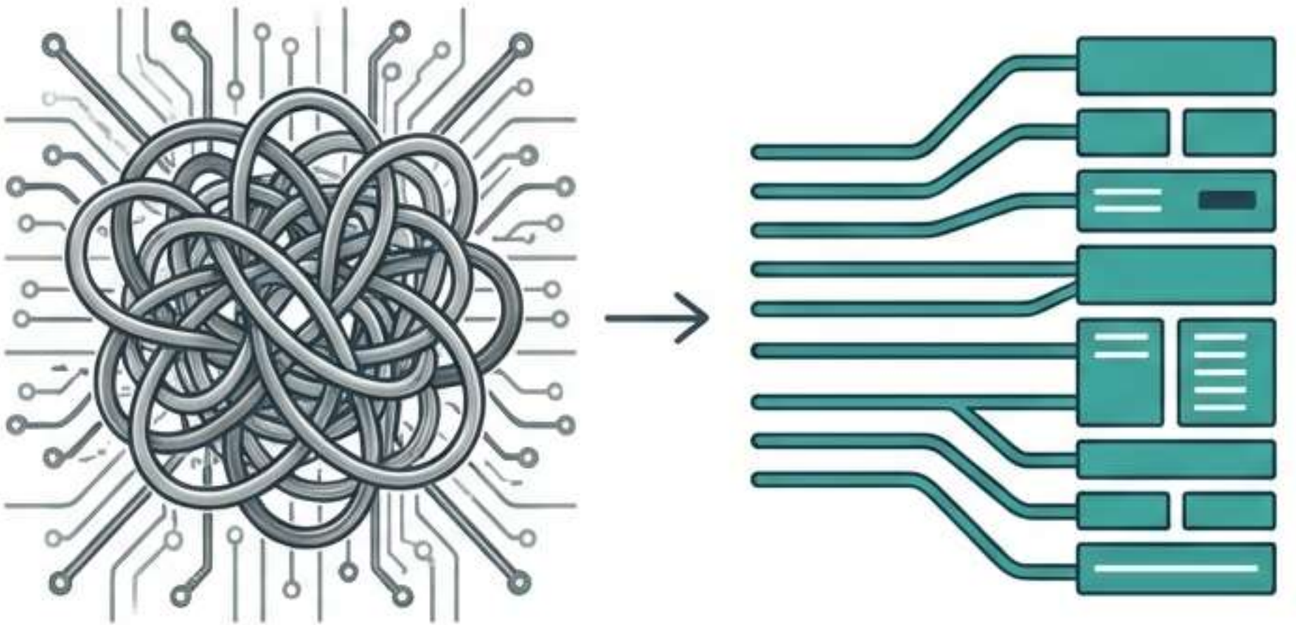
Prioritize automating Regression tests; leave exploratory work for Manual.

Environment Parity

Mirror production environments to catch compatibility bugs.

Architecture & Implementation Challenges

- **Flaky Tests:** Automated scripts that fail inconsistently due to timing/environment.
- **Maintenance Overhead:** Cost of updating scripts as UI changes.
- **Integration Complexity:** Debugging failures across interacting modules.



Spaghetti Code/Complex Dependencies

Modularized & Robust Architecture

Scalability & Performance Constraints

- **Load Management:** Stability during volume spikes.
- **Stress Points:** Identifying breaking points under extreme conditions.
- **Device Fragmentation:** Consistency across OS versions.

Interview Preparation: Key Concepts

Top Concept: Smoke vs. Sanity Testing.
Talking Point: Smoke is general build stability; Sanity is deep, targeted checking after a fix.

Scenario Question: "We just released a hotfix for the checkout button. What test do you run?"
Answer: Sanity testing on the checkout flow, followed by a targeted regression.

Insight: Interviewers look for understanding of resource optimization—knowing when to test deep vs. wide.

Session Key Takeaways

Testleaf
Always Ahead

Testing is Layered

From Unit (code) to System (behaviour).

Automation supports Manual

It handles repetition (Regression), not replacement.

Terminology Matters

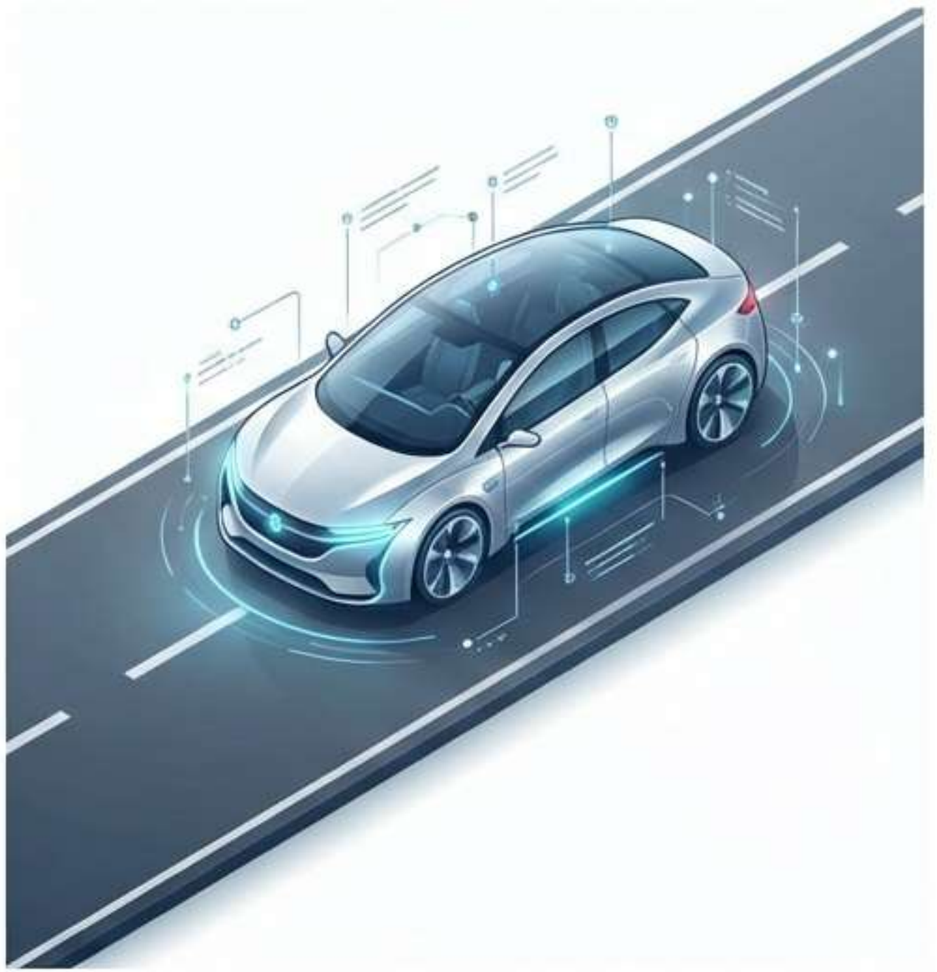
Smoke (General Health) vs. Sanity (Targeted Fix) vs. Regression (Safety Net).

Non-Functional is Critical

Security, Load, and Compatibility are vital for success.

Executive Summary

- Software testing is a disciplined process ensuring code quality, security, and performance.
- Leveraging Manual and Automated strategies alongside White and Black box methodologies delivers robust software.
- Mastery of workflows (Smoke -> Sanity -> Regression) ensures efficient release cycles.



Knowledge Check: Question 1

Question: Which type of testing is performed to ensure the basic functionality of an application is stable immediately after a new build?

A	Regression Testing
B	Smoke Testing
C	Load Testing
D	Compatibility Testing

Knowledge Check: Question 2

Question: Which testing is typically performed after minor code changes to verify that specific functionality is working as expected?

- **A** Smoke Testing
- **B** Sanity Testing
- **C** Regression Testing
- **D** Stress Testing

Testleaf
Always Ahead

Knowledge Check: Question 3

Question: Which testing type aims to identify system stability and breaking points under extreme conditions?

- **A** Performance Testing
- **B** Security Testing
- **C** Stress Testing
- **D** Sanity Testing

Knowledge Check: Question 4

Question: What type of testing evaluates system responsiveness, speed, and scalability under expected workloads?

- **A** Smoke Testing
- **B** Regression Testing
- **C** Performance Testing
- **D** Compatibility Testing

Testleaf
Always Ahead

Knowledge Check: Question 5

Question: Which testing ensures that the application works correctly across different devices, browsers, and operating systems?

A Compatibility Testing

B Regression Testing

C Accessibility Testing

D Load Testing

Testleaf
Always Ahead

Assessment Answer Key

Q1 – B (Smoke Testing) ✔

Q2 – B (Sanity Testing) ✔

Q3 – C (Stress Testing) ✔

Q4 – C (Performance Testing) ✔

Q5 – A (Compatibility Testing) ✔