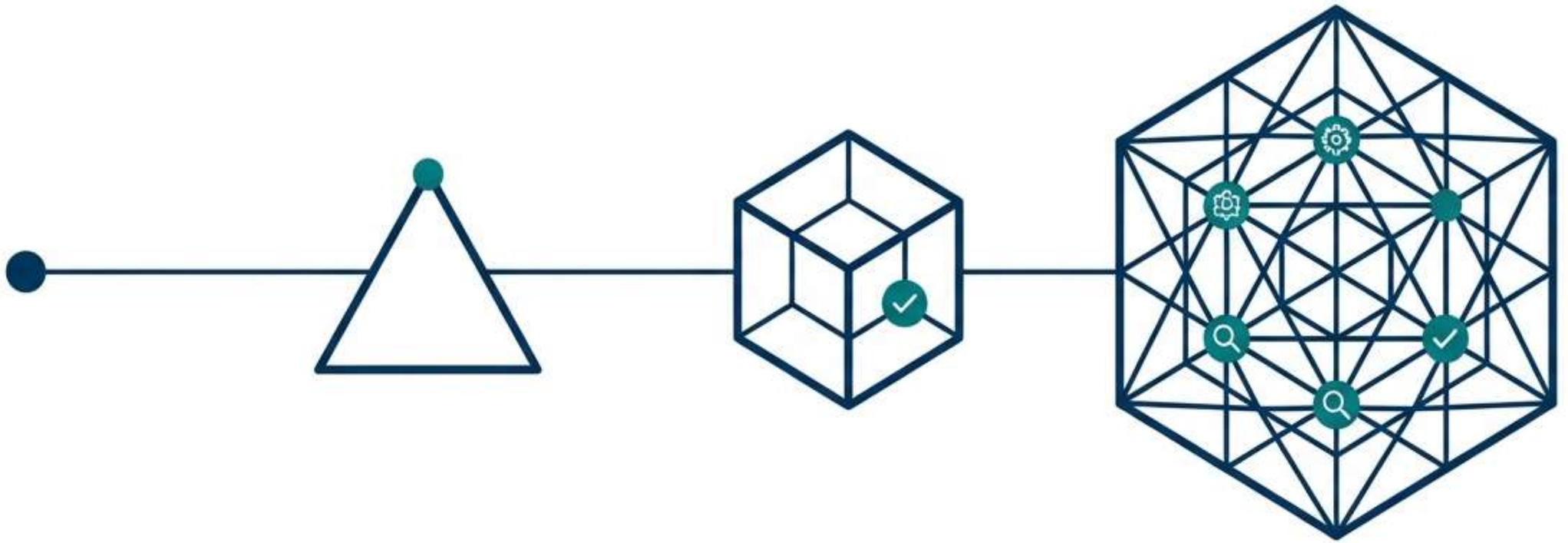
Strategic Software Testing Levels

Testleaf
Always Ahead

From Unit Verification to User Acceptance



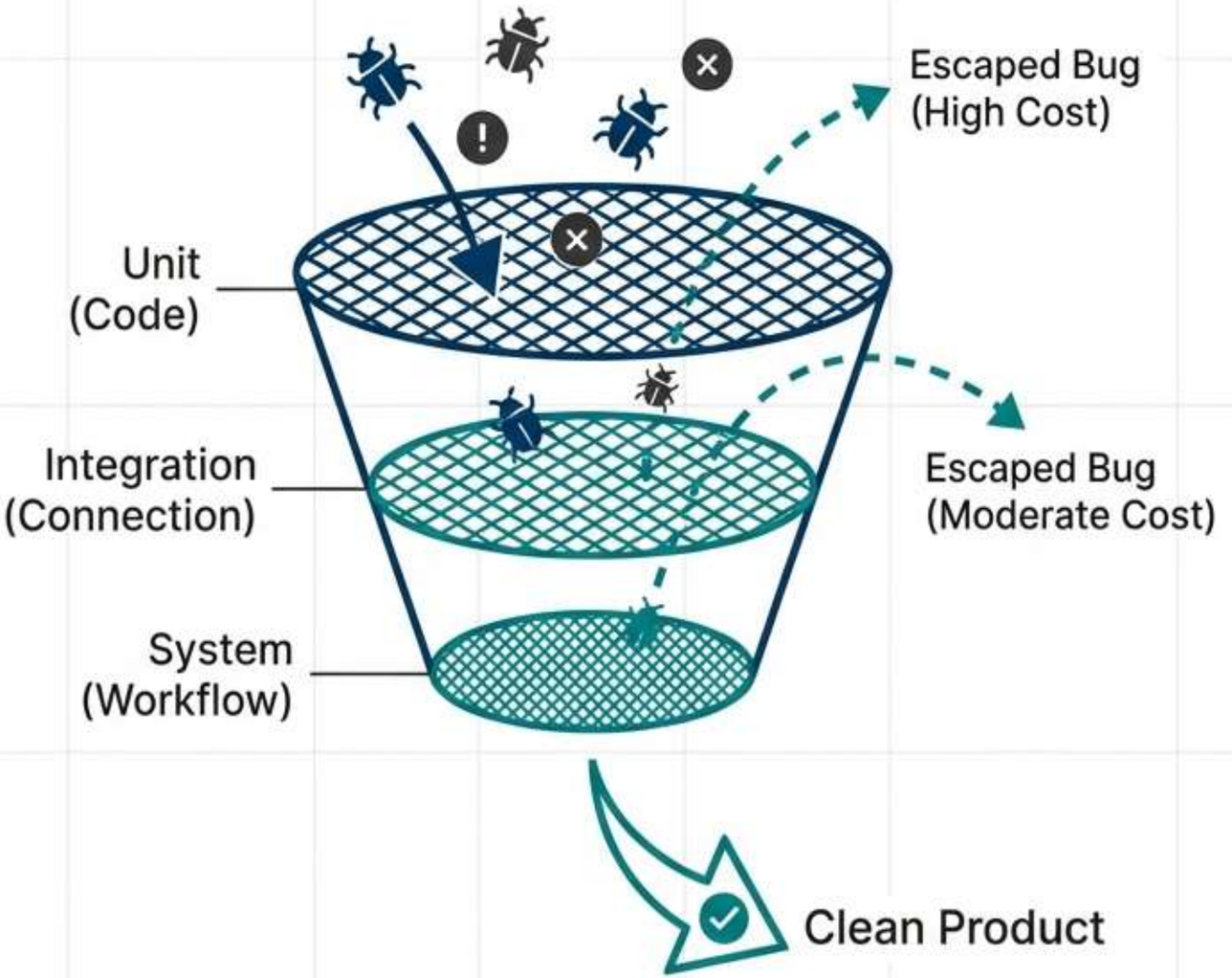
UNIT TESTING → **INTEGRATION TESTING** → **SYSTEM TESTING** → **USER ACCEPTANCE TESTING**

Component Isolation | Interface & Interaction | End-to-End Functionality | Validation against Requirements

The Philosophy of Layered Testing

Testleaf
Always Ahead

The Cost of Quality: Testing is a progressive filtration process. Identifying bugs during early development saves significant time and budget compared to post-release fixes.



The Four Pillars of Software Testing

Testleaf
Always Ahead

Unit Testing (White Box)

Test individual component

Integration Testing (Grey Box)

Test components work together

System Testing (Black Box)

Test the entire system

Acceptance Testing (User-Centric)

Test the final system

Micro-Level Verification: Unit & Integration

Unit Testing (White Box)

Focus: Smallest testable part of the system.
Owner: Developers.
Benefit: Enables rapid changes and documentation.

Integration Testing (Grey Box)

Focus: Data connectivity between modules.
Owner: Joint effort (Dev + Test).
Goal: Verifying communication.

Macro-Level Validation: System & Acceptance

Testleaf
Always Ahead

System Testing (Black Box)

Scope: Validating fully integrated application against specifications.
Owner: Independent QA Testers.
Goal: End-to-end compliance.

Acceptance Testing (User-Centric)

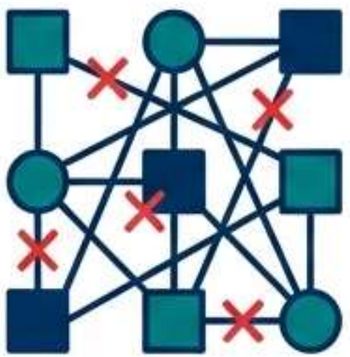
Scope: Final validation for business needs.
Alpha Phase: Developer environment.
Beta Phase: End-user environment.

Strategic Integration Approaches

Testleaf
Always Ahead

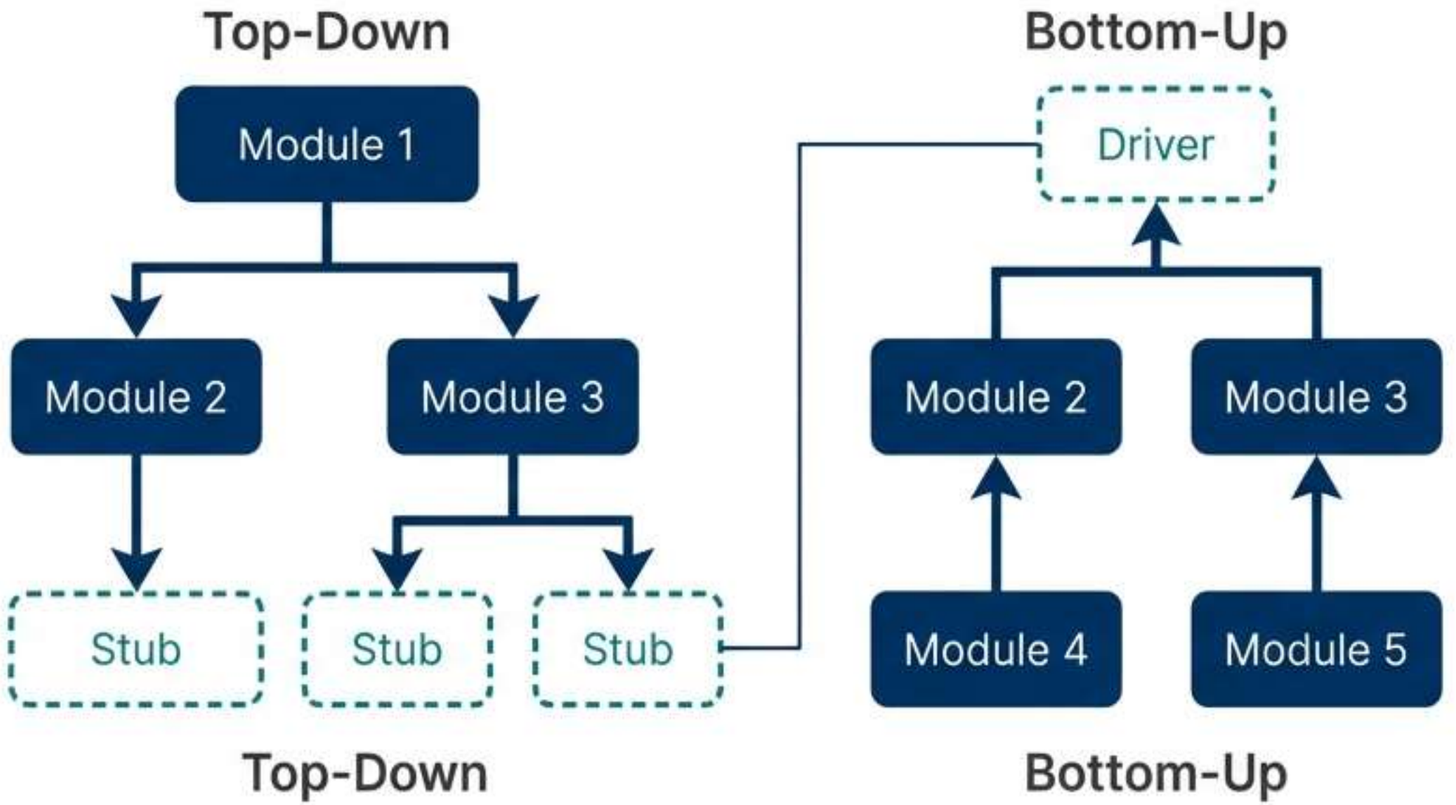
Big Bang Approach

Integrating all modules simultaneousy. High risk, difficult fault localisation.



Incremental Approach

Integrating logically related modules gradually.



Real-World Application: E-Commerce Platform

Testleaf
Always Ahead

Unit

Verify Tax Calculation Function

Integration

Cart sends total to Payment Gateway

System

Full Checkout: Search > Cart > Pay > Email

Acceptance

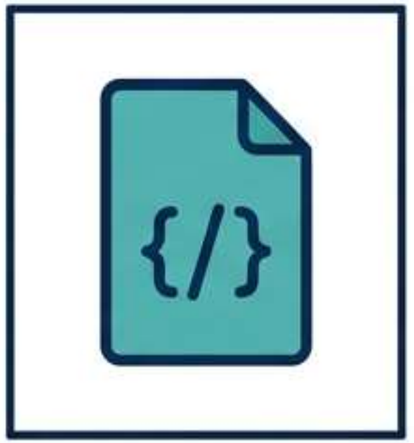
Beta testers use app during holiday sale

Guidelines for Effective Implementation

Testleaf
Always Ahead

Shift Left Strategy

Prioritise Unit Testing to catch errors when they are cheapest to fix.



Live Documentation

Treat Unit Tests as documentation to aid future developers.

Incremental Integration

Avoid Big Bang testing to ensure easier fault localisation.

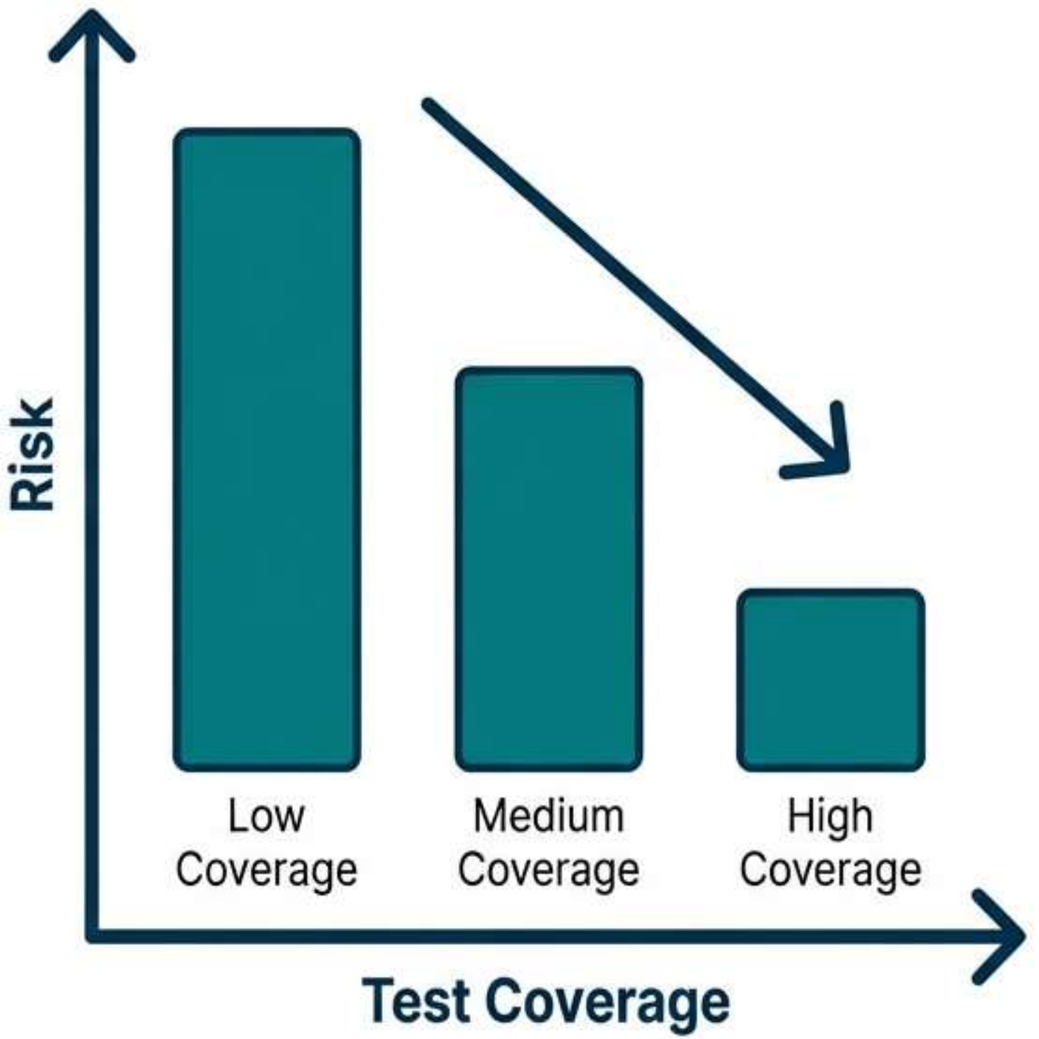
Environment Parity

Ensure Acceptance testing mirrors the production environment.

Critical Learning Points

Testleaf
Always Ahead

- **Progression**: Testing moves from granular (Unit) to holistic (System).
- **Responsibility**: Ownership transitions from Developers to QA to Users.
- **Strategy**: Incremental integration offers superior control over Big Bang.
- **Value**: Rigorous lower-level testing reduces production risks and costs.



Session Summary

Testleaf
Always Ahead

- We defined Unit, Integration, System, and Acceptance testing.
- We analysed why Incremental Integration is preferred over Big Bang.
- We distinguished between Alpha (Dev) and Beta (User) environments.