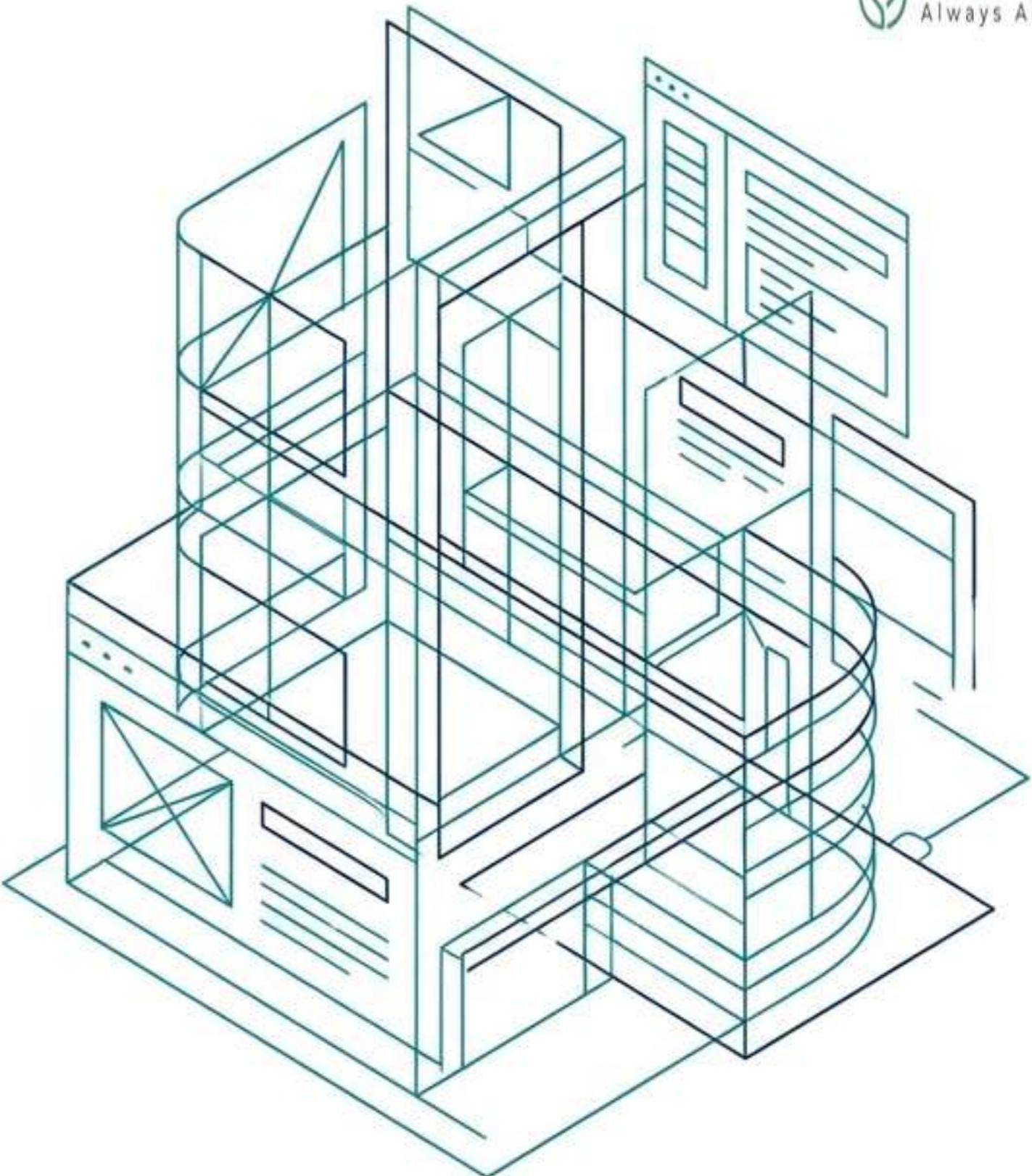


Fundamentals of Software Quality Assurance

Strategies, Processes, and Best
Practices for Professional Testing



The Strategic Value of Software Testing

Definition: A systematic process of evaluating software to verify adherence to customer requirements and identify defects before deployment.

Risk Mitigation: Identifies issues prior to launch to prevent costly post-deployment failures.

Brand Reputation: Ensures only quality products reach the consumer, directly elevating customer satisfaction and trust.

Quality Control: The process ensures the final deliverable matches the initial specifications.



Core Principles of Quality Assurance



The Golden Rule: Testing is not an afterthought—it acts as the gatekeeper, verifying the product does exactly what it is supposed to do.

Methodologies: Manual vs. Automation



Manual Testing

Human-driven verification of features.

Simulates real-world user behaviour directly.

Essential for usability and non-repetitive scenarios.



Automation Testing

Utilises tools (specifically Selenium WebDriver) to execute scripted tests.

Efficiency: Reduces execution time for repetitive tasks.

Reusability: Write a script in one language and run it across multiple browser platforms.

The Testing Lifecycle: When to Start and Stop



Testing in Practice: Distinguishing Roles

	 Testing (QA Professional)	 Debugging (Developer)
Goal	 Identification	 Resolution
Process	 Executing application to find defects without correcting them.	 Identifying, isolating, and fixing errors in the code.
The Question	 What is wrong?	 Why is it wrong?

Quality Assurance Best Practices



Start Early.

Do not wait for full development. Reviewing requirements is a critical form of testing.



Define 'Done'.

Establish clear exit criteria (deadlines, bug rates) to avoid aimless testing cycles.



Balanced Approach.

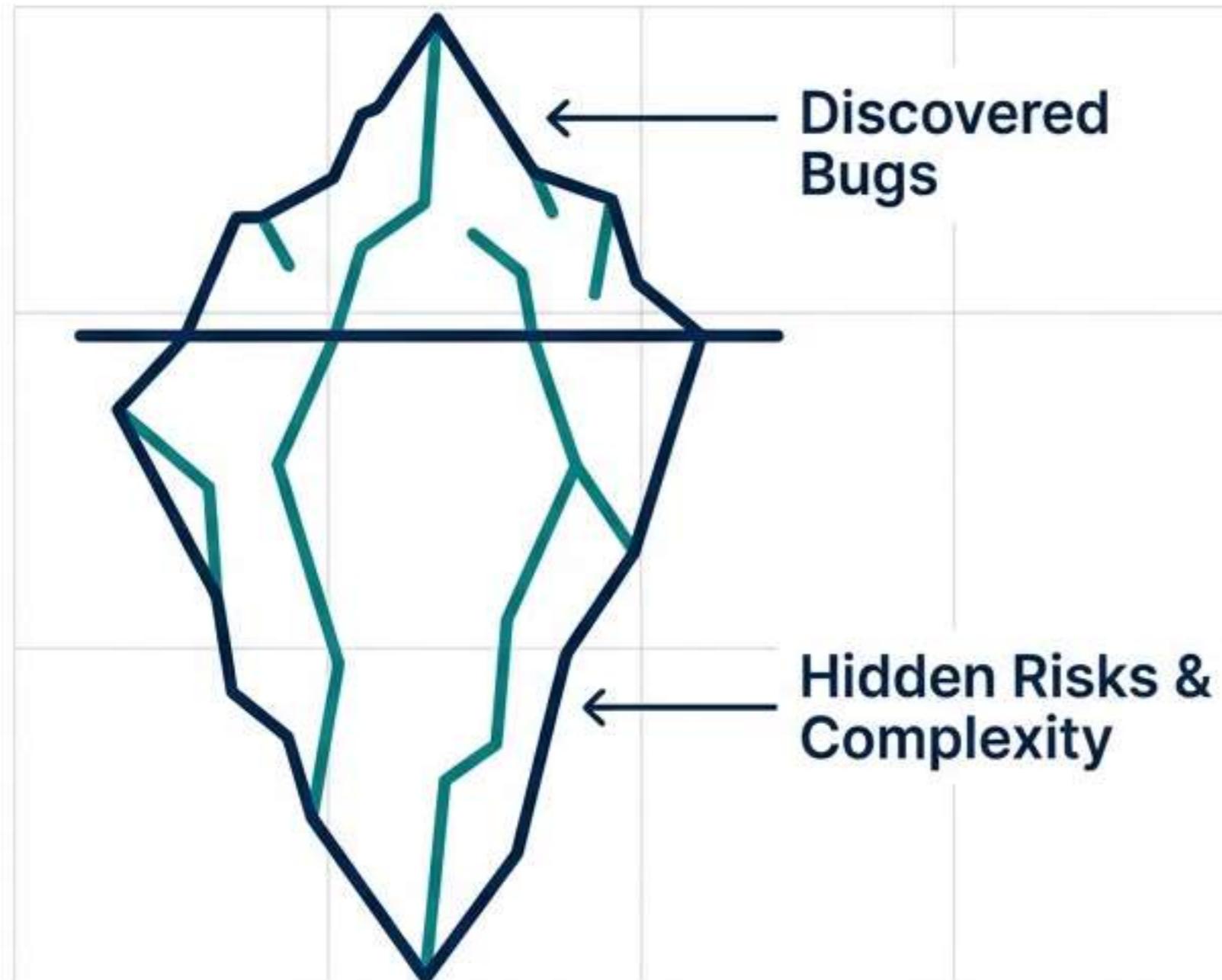
Automation is powerful for regression, but do not replace manual intuition for UX scenarios.



Shared Responsibility.

Quality is a team effort. Testers identify bugs; the entire team owns the product quality.

Common Pitfalls and Challenges



1. The '100% Defect-Free' Fallacy:

Testing significantly reduces risk but cannot logically prove the absence of all defects.



2. Automation Over-Reliance:

Automation is for efficiency in repetition. It cannot replace human insight in exploratory or UX testing.



3. The Blame Culture:

Defects often stem from ambiguous requirements or complex code integration; quality is a collective ownership, not just the tester's fault.



Career Preparation: Interview Insights



Testing vs. Debugging.

Focus your answer on 'Identification' (QA) versus 'Resolution' (Dev).



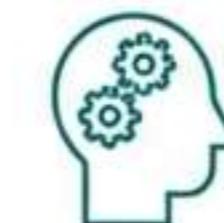
Automation Strategy.

Explain that you automate for reusability and platform coverage (Selenium), not to replace the manual tester's intuition.



When to Start?

Emphasise the 'Shift Left' approach—verifying requirements documents before coding begins.



Professional Mindset.

Reject the myth that 'anyone can test.' Highlight specific analytical skills and structured methodologies.

Summary of Key Concepts

Definition

Testing validates requirements and identifies defects to ensure customer satisfaction.



Methodology

A hybrid of Manual (human-centric) and Automation (script-centric/Selenium) is optimal.



Timing

Testing is continuous—starting with requirement reviews and ending only when exit criteria are met.



Roles

Testers identify issues; Developers fix them. Quality is a collective team effort.

