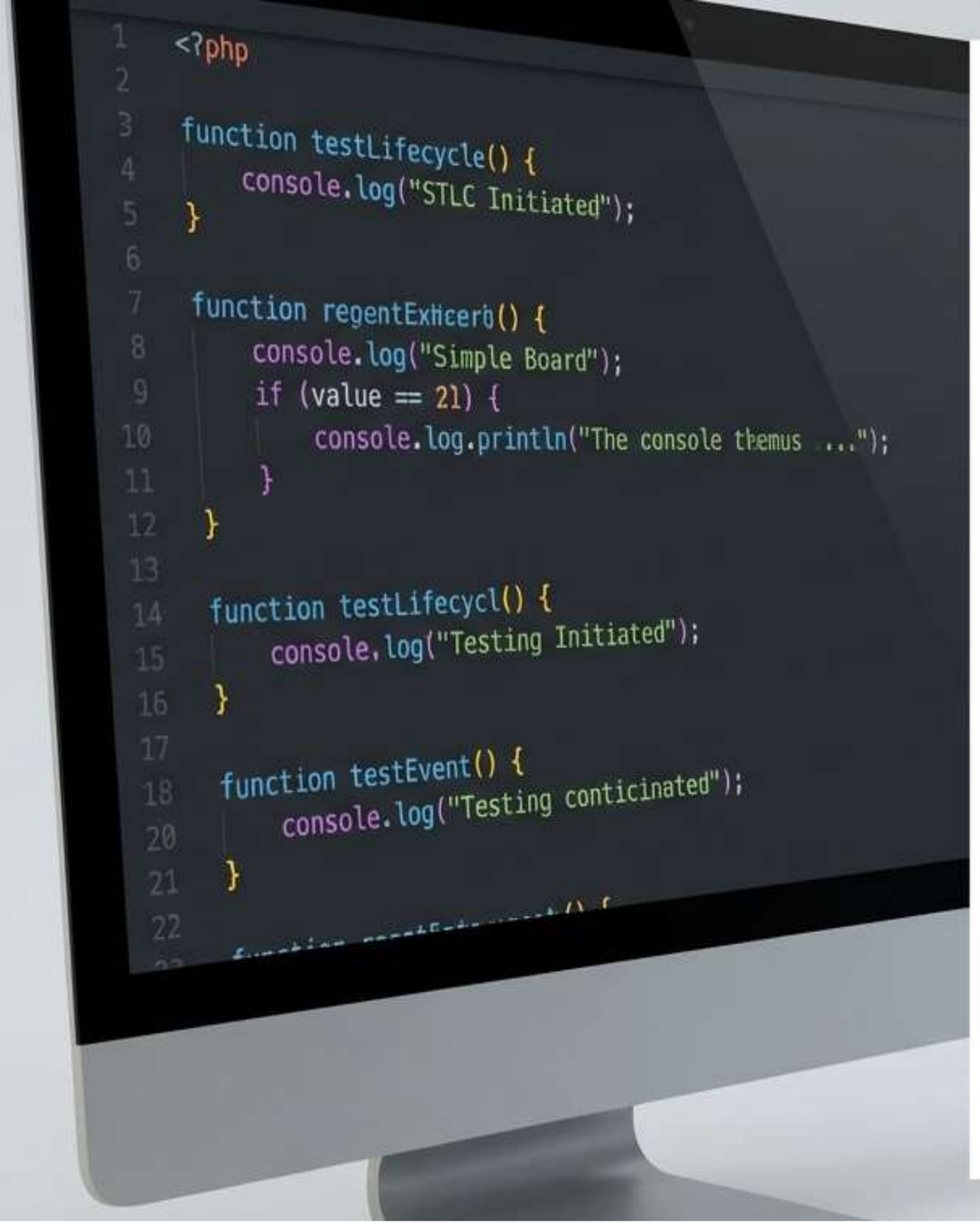


Testleaf
Always Ahead

The Software Testing Life Cycle (STLC)

Structured Frameworks for Ensuring Software Quality and Reliability

Defining the STLC

What is STLC?

A sequence of specific activities conducted during the testing process to ensure software quality goals are met. It shifts the focus from ad-hoc checking to a systematic quality assurance process.

Why is it Critical?

- **Standardisation**: Provides a consistent step-by-step process.
- **Efficiency**: Enables script reuse across platforms (e.g., Selenium).
- **Quality Assurance**: Reduces defect leakage by validating quality at every step.

The Six Phases of STLC

Testleaf
Always Ahead

The Golden Rule: Testing is not a single event; it is a cycle.

Phases 1–3: The Planning Stage

Testleaf
Always Ahead

1. Requirement Analysis

- **Activity:** Scrutinise documents to identify testable requirements. Clarify doubts with stakeholders (**Entry Criteria**).
- **Deliverables:** Requirement Traceability Matrix (RTM) & Automation Feasibility Report.

2. Test Planning

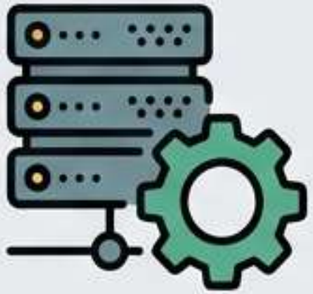
- **Activity:** QA Lead defines strategy, estimates effort, selects tools, and assigns roles.
- **Deliverables:** Test Plan Document & Test Strategy Document.

3. Test Case Development

- **Activity:** Creation and verification of scripts, identification of test data, and baselining test cases.
- **Deliverables:** Verified Test Cases / Scripts & Test Data.

Phases 4–6: Execution & Closure

Testleaf
Always Ahead



4. Test Environment Setup

- **Activity:** Hardware/software provisioning and network configuration.
- **Critical Step:** Perform a 'Smoke Test' to ensure environment readiness.

5. Test Execution

- **Activity:** Execute scripts, log defects, map to RTM, and retest fixes.
- **Deliverables:** Completed RTM, Defect Reports, Updated Test Results.

6. Test Cycle Closure

- **Activity:** Analyse artifacts, document lessons learned, and evaluate coverage.
- **Deliverables:** Test Closure / Summary Report.

The Deliverables Chain

Testleaf
Always Ahead

Requirements → Yields →  → Yields → Test Plan Document → Yields → Test Scripts & Data → Yields → Defect Reports → Yields → 

Traceability Matrix (RTM)

Test Summary Report

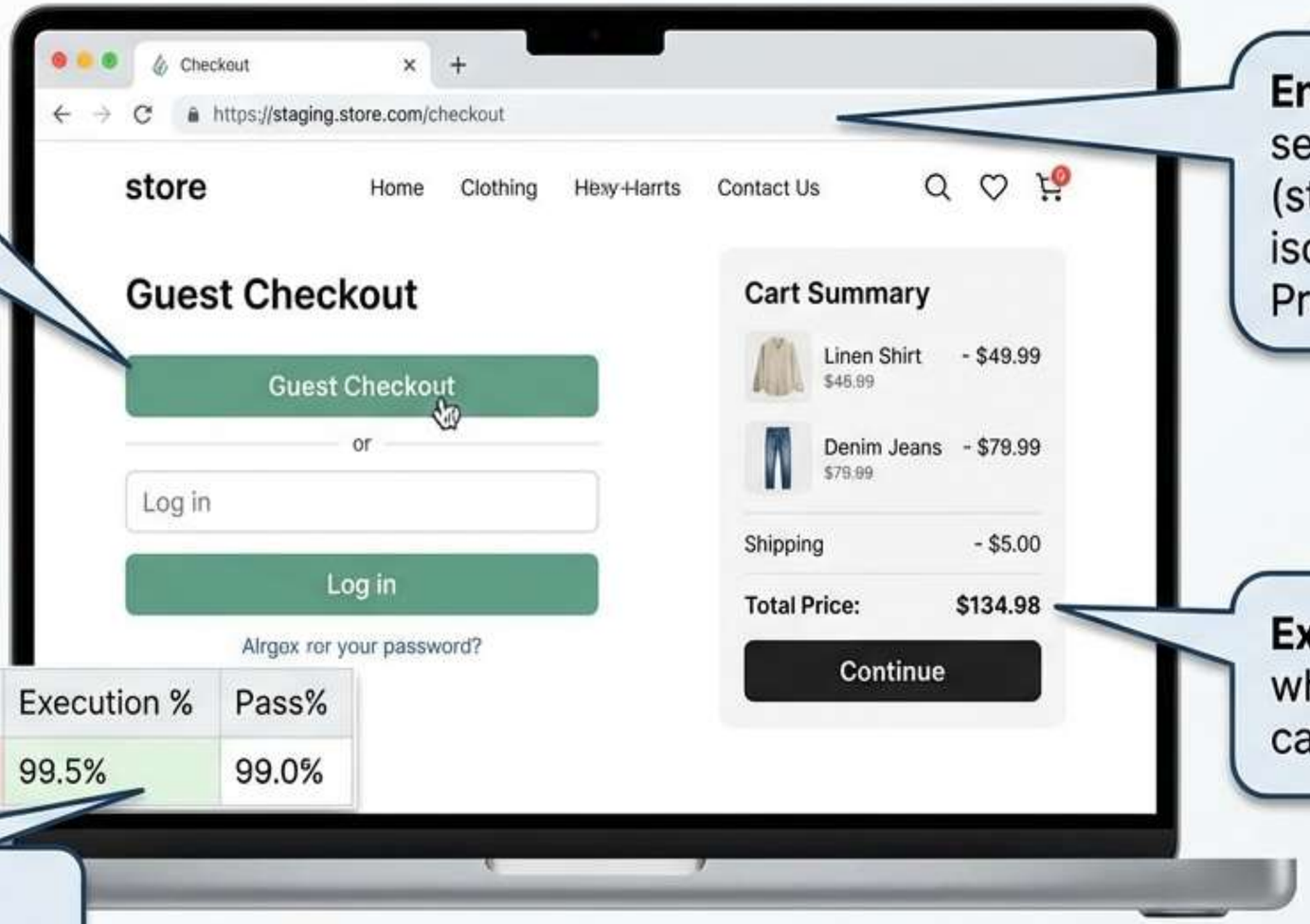
Applied STLC: E-Commerce Scenario

Testleaf
Always Ahead

- Context: Testing a new ‘Guest Checkout’ feature (Release 1.2)

Analysis: Identified ‘Guest User’ as a distinct testable requirement (BID001).

Environment: Staging server (staging.store.com) isolated from Production.



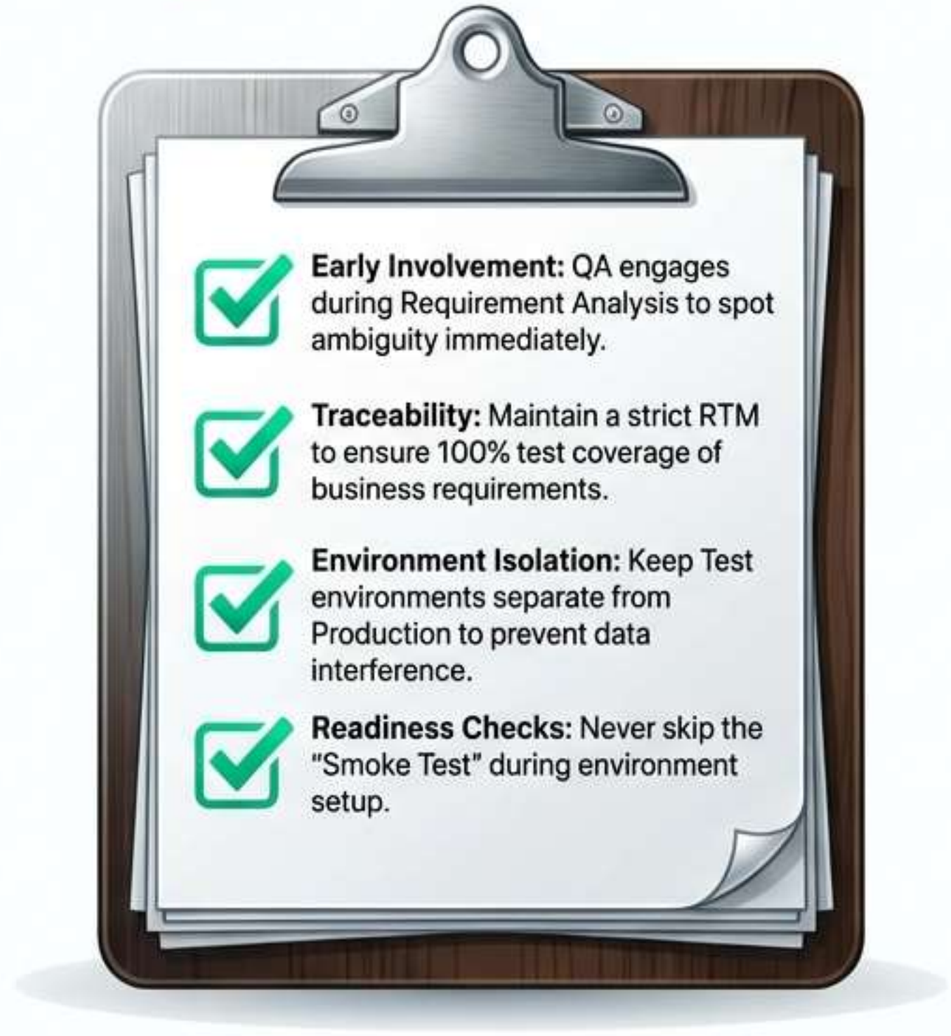
Total Test Cases	Passed	Failed	Execution %	Pass%
880	871	5	99.5%	99.0%

Execution: Found bug where shipping cost calculated incorrectly.

Closure: 99.5% Execution Complete, 99.0% Pass Rate.

Testleaf
Always Ahead

Ensuring Excellence in STLC



Testleaf
Always Ahead

Navigating Implementation Challenges

Environment Drift

- **Issue:** Test environment differs from production (missing libraries).
- **Solution:** Use Infrastructure as Code (IaC) to standardise provisioning.

Automation Feasibility

- **Issue:** Attempting to automate unstable features.
- **Solution:** Use a Feasibility Analysis Score. If Total Score < 50, do not automate.

Factors	Total Score	Score / Lines of Business			
		AUT	O	0	0
Technical Analysis	20	0	0	0	0
Complexity	15	0	0	0	0
Aoplicatioo Stability	15	0	0	0	0
Test Data	15	0	0	0	0
Application Size	10	0	0	0	0
Reusability of Automsted Scripts	10	0	0	0	0
Execution acress Environments	15	0	0	0	0
TOTAL SCORE	100	0	0	Application not suitable fer Automatien	

Ambiguous Requirements

- **Issue:** Vague needs lead to wasted scripting time.
- **Solution:** Mandatory stakeholder sign-off on acceptance criteria.

Troubleshooting

- **Issue:** Intermittent script failures.
- **Solution:** Implement robust logging and screenshot-on-fail mechanisms.

Summary of Key Insights

STLC is a systematic, six-phase cycle, not a single event.

Planning (Phases 1-3) is just as critical as Execution (Phases 4-6).

The Requirement Traceability Matrix (RTM) is the central link between business needs and verification.

Test Closure is vital for process improvement—analysing ‘severity distribution’ drives future efficiency.