

Test Strategy Document

Tricentis Demo Web Shop

Document Version: 1.0

Prepared By: Lakindu De Silva

Date: December 15, 2025

Project: Software Quality Engineering Coursework

1. Introduction

1.1 Purpose

This Test Strategy document outlines the testing approach, methodologies, and resources required to ensure the quality of the Tricentis Demo Web Shop application. It provides a high-level overview of the testing activities that will be performed.

1.2 Scope

This strategy covers:

- Manual Testing (40 test cases)
- Automated Testing (4-5 critical test cases)
- Defect Management
- Test Reporting

1.3 Objectives

- Validate all core functionalities of the web shop
 - Identify and document defects
 - Ensure usability and user experience standards are met
 - Verify form validations and boundary conditions
 - Provide release readiness recommendation
-

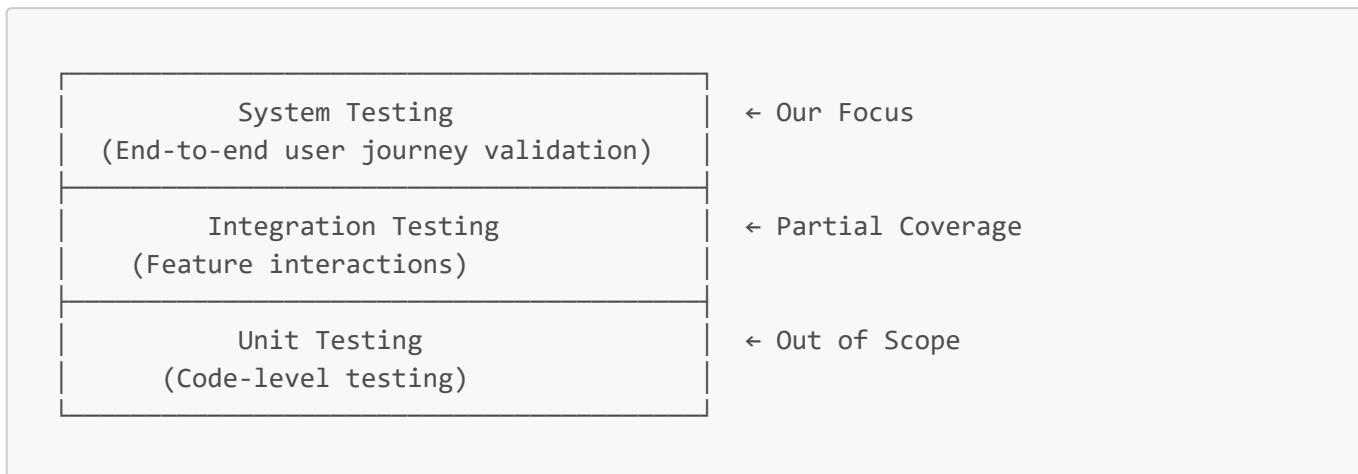
2. Test Approach

2.1 Testing Types

Testing Type	Description	Coverage
Functional Testing	Verify features work as expected	50% of test cases
Usability Testing	Assess user-friendliness	15% of test cases
UI/UX Testing	Verify visual elements and layout	15% of test cases
Boundary Testing	Test edge cases and limits	10% of test cases

Testing Type	Description	Coverage
Negative Testing	Test with invalid inputs	10% of test cases

2.2 Testing Levels



2.3 Test Design Techniques

Technique	Application
Equivalence Partitioning	Form inputs (valid/invalid data groups)
Boundary Value Analysis	Quantity fields, password length
Decision Table	Login combinations, checkout paths
Use Case Testing	End-to-end user journeys
Error Guessing	Based on common web app issues
Exploratory Testing	Ad-hoc discovery of issues

3. Test Environment

3.1 Hardware Requirements

- Computer with minimum 4GB RAM
- Stable internet connection
- Screen resolution: 1920x1080 (minimum 1366x768)

3.2 Software Requirements

Component	Specification
Operating System	Windows 10/11
Primary Browser	Google Chrome (Latest) / Microsoft Edge
Automation Tool	Playwright with JavaScript

Component	Specification
IDE	Visual Studio Code
Screenshot Tool	Windows Snipping Tool / Built-in

3.3 Test URL

- **Production Test Environment:** <https://demowebshop.tricentis.com/>
-

4. Test Automation Strategy

4.1 Automation Scope

Automated Test Cases (4-5 Critical Scenarios):

TC ID	Scenario	Priority	Automation Reason
AT-001	User Registration	Critical	Repetitive, data-driven
AT-002	User Login	Critical	High frequency, regression
AT-003	Add Product to Cart	Critical	Core functionality
AT-004	End-to-End Checkout	Critical	Business critical flow

4.2 Automation Tool Selection

Selected Tool: Playwright

Criteria	Playwright Score
Easy Setup	<input checked="" type="checkbox"/> Excellent
Cross-browser Support	<input checked="" type="checkbox"/> Chrome, Firefox, Safari
Modern Web Support	<input checked="" type="checkbox"/> Excellent
Documentation	<input checked="" type="checkbox"/> Comprehensive
Community Support	<input checked="" type="checkbox"/> Active
Reporting	<input checked="" type="checkbox"/> Built-in HTML reports

4.3 Automation Framework Structure

```
automation/
└── tests/
    ├── registration.spec.js
    ├── login.spec.js
    ├── cart.spec.js
    └── checkout.spec.js
└── pages/                                (Page Object Model - optional)
```

```
└── test-data/
    └── users.json
└── reports/
└── screenshots/
```

5. Test Data Management

5.1 Test Data Categories

Category	Examples	Source
Valid Data	Correct email, passwords	Generated
Invalid Data	Wrong format, empty fields	Defined
Boundary Data	Min/max lengths	Calculated
Existing Data	Duplicate emails	Pre-created

5.2 Sample Test Data

User Registration:

Valid User:

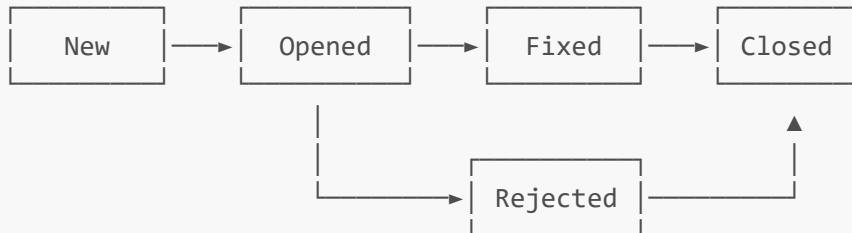
- First Name: Test
- Last Name: User
- Email: testuser_[timestamp]@test.com
- Password: Test@123

Invalid User:

- Email: invalid-email
- Password: 123 (too short)

6. Defect Management

6.1 Defect Lifecycle



6.2 Defect Severity Levels

Severity	Description	Example
Critical	System crash, data loss, no workaround	Checkout fails completely
High	Major feature broken, workaround exists	Cannot add to cart from listing
Medium	Feature partially working	Sort by price not working
Low	Minor issue, cosmetic	Typo in label

6.3 Defect Priority Levels

Priority	Description	Resolution Time
P1 - Urgent	Must fix immediately	Same day
P2 - High	Fix before release	1-2 days
P3 - Medium	Fix if time permits	Next release
P4 - Low	Nice to have	Backlog

7. Entry and Exit Criteria

7.1 Entry Criteria

- ☐ Test environment is accessible
- ☐ Test cases are documented and reviewed
- ☐ Test data is prepared
- ☐ Automation scripts are ready
- ☐ Testing tools are configured

7.2 Exit Criteria

- ☐ All planned test cases executed
- ☐ All Critical and High severity defects resolved (or accepted)
- ☐ Test coverage meets target (>95%)
- ☐ Automation tests pass successfully
- ☐ Test reports generated and reviewed

8. Risk Assessment

Risk ID	Risk Description	Impact	Probability	Mitigation
R001	Website unavailable	High	Low	Use local recordings/screenshots
R002	Test data conflicts	Medium	Medium	Use unique timestamps
R003	Time constraints	High	Medium	Prioritize critical tests
R004	Browser compatibility	Medium	Low	Focus on primary browser

Risk ID	Risk Description	Impact	Probability	Mitigation
R005	Automation flakiness	Medium	Medium	Add proper waits/retries

9. Deliverables

Deliverable	Description	Format
Test Cases Document	40 manual test cases	Markdown/Excel
Automation Scripts	4-5 Playwright tests	JavaScript
Defect Report	Logged defects	Markdown
Test Execution Report	Pass/Fail results	Markdown
Test Coverage Report	Coverage metrics	Markdown
Final Report	Complete assessment	PDF
Presentation	Summary slides	PowerPoint

10. Roles and Responsibilities

Role	Name	Responsibilities
QA Engineer	Lakindu De Silva	Test design, execution, automation, reporting

11. Schedule

Phase	Activities	Duration
Phase 1	Documentation & Test Design	30 min
Phase 2	Test Automation	45 min
Phase 3	Manual Testing & Reporting	2-3 hours

12. Approval

Role	Name	Signature	Date
QA Engineer	Lakindu De Silva	_	-
Reviewer	_	-	-