**Agile Test Strategy Document – NopCommerce Website**

**1. Revision History**

| **Date** | **Version** | **Author** | **Description** |
| --- | --- | --- | --- |
| 26-Aug-2025 | 1.0 | Team A | Initial draft for Agile Test Strategy |
| 28-Aug-2025 | 1.1 | Team A | Updated with NopCommerce-specific scope & approach |
| 30-Aug-2025 | 1.2 | Team A | Finalized for project submission |

**Table of Contents**

1. Revision History
2. Introduction
3. Objectives
4. Scope of Testing
5. Testing Approach
6. Sprint Plan
7. Types of Testing
8. Entry and Exit Criteria
9. Test Deliverables
10. Tools and Resources
11. Roles and Responsibilities
12. Risks and Mitigation

**2. Introduction**

This document defines the **Agile Test Strategy** for the **NopCommerce demo website** (<https://demo.nopcommerce.com/>). It describes the scope, objectives, approach, tools, and responsibilities for ensuring high-quality deliverables within Agile sprints.

**3. Objectives**

* Deliver a **tested, working increment** at the end of each sprint.
* Detect and resolve defects **early** through continuous testing.
* Maximize **test coverage** using manual + automated testing.
* Ensure **traceability** of test cases to user stories in Jira.
* Support Agile ceremonies: sprint planning, daily standups, retrospectives.

**4. Scope of Testing**

Testing covers core modules of the NopCommerce demo site:

* **User Registration & Login**
* **Product Search & Filters**
* **Shopping Cart & Wishlist**
* **Checkout & Payment Integration**
* **Order History & Account Management**
* **UI Components (dropdowns, checkboxes, radio buttons, images)**
* **Cross-browser compatibility (Chrome, Firefox, Edge)**

Out of Scope:

* Backend database performance testing
* API testing beyond UI workflow
* Security penetration testing

**5. Testing Approach**

* **Agile Scrum methodology** will be followed.
* Testing will be **iterative & continuous** in alignment with sprint cycles.
* Test Scenarios derived from **user stories & acceptance criteria**.
* **TestNG + Cucumber BDD framework** for automation.
* **Page Object Model (POM)** design for maintainability.
* **Extent Reports + Screenshots** for reporting.
* Manual testing for exploratory + negative cases; automation for regression & smoke packs.

**6. Sprint Plan**

| **Sprint** | **Duration** | **Focus Areas** | **QA Activities** |
| --- | --- | --- | --- |
| Sprint 1 | Aug 28 – Aug 31 | - User Registration & Login  - Basic Navigation | - Functional Testing  - Smoke Testing  - UI Validation |
| Sprint 2 | Sep 1 – Sep 3 | - Product Search & Category Browsing  - Wishlist/Compare | - Integration Testing  - Boundary Testing |
| Sprint 3 | Sep 4 – Sep 6 | - Shopping Cart  - Checkout & Payments | - Regression Testing  - Negative Testing  - Security validation |
| Sprint 4 | Sep 7 – Sep 9 | - Order History  - My Account Features  - Cross-browser checks | - System Testing  - UAT  - Performance Validation |

**7. Types of Testing**

* **Smoke Testing**: Verify basic workflows (login, navigation).
* **Functional Testing**: Validate features per acceptance criteria.
* **Regression Testing**: Ensure no existing functionality breaks.
* **Boundary Testing**: Validate input limits (e.g., qty in cart).
* **Integration Testing**: Test interaction between cart, checkout, and payment.
* **UAT (User Acceptance Testing)**: Validate business workflows end-to-end.
* **Performance Testing**: Ensure page load & responsiveness under normal load.
* **Security Testing (basic)**: Input validation for login/search to avoid SQLi/XSS.

**8. Entry and Exit Criteria**

**Entry Criteria:**

* User stories approved with acceptance criteria.
* Development completed for sprint backlog items.
* QA environment deployed with test data.

**Exit Criteria:**

* All P1 & P2 defects resolved/closed.
* Minimum 95% test case execution completed.
* Regression pack executed successfully.
* Sprint QA report submitted.

**9. Test Deliverables**

* Test Plan
* Test Scenarios & Test Cases (Excel / Jira)
* Automated Scripts (Java + Selenium + TestNG + Cucumber)
* Extent HTML Reports with Screenshots
* Sprint-wise QA Summary Report
* Defect Reports in Jira

**10. Tools and Resources**

* **Test Management**: Jira (stories, tasks, bugs)
* **Automation**: Java + Selenium + TestNG + Cucumber
* **Reporting**: Extent Reports
* **CI/CD**: Jenkins (optional)
* **Data Management**: Excel / JDBC integration
* **Communication**: Slack, MS Teams, Email

**11. Roles and Responsibilities**

| **Role** | **Responsibility** |
| --- | --- |
| **QA Lead** | Define test strategy, review deliverables, assign tasks |
| **QA Engineer** | Write & execute test cases, develop automation scripts |
| **Scrum Master** | Facilitate Agile ceremonies, remove blockers |
| **Developer** | Fix bugs, support QA during testing |
| **Product Owner** | Define user stories, prioritize backlog, accept features |

**12. Risks and Mitigation**

| **Risk** | **Mitigation** |
| --- | --- |
| Delay in Dev Deliverables | Early sync with devs, buffer in sprint planning |
| Environment Instability | Use stable QA environment, fallback test data |
| High Defect Leakage | Run smoke/regression before each build sign-off |
| Incomplete Acceptance Criteria | Engage in backlog grooming & PO clarification |
| Parallel Team Conflicts | Use Git branching strategy & CI builds |