

|  |
| --- |
|  |
|  |

|  |  |
| --- | --- |
| Acronyms & Abbreviations | |
| QMS | Quality Management System |
| POC | Proof of Concept |
| MVP | Minimum Viable Product |
| SOW | Statement of Work |
| DM/PM | Delivery Manager/Project Manager |

Contents

[1 INtroduction to Project and Project Specific Impact to Testing 4](#_Toc194339158)

[2 Scope of Testing 4](#_Toc194339159)

[2.1 In Scope 4](#_Toc194339160)

[2.2 Out of Scope 4](#_Toc194339161)

[2.3 Third-Party Systems 5](#_Toc194339162)

[3 Quality & Acceptance Criteria & Critical Success Factors 5](#_Toc194339163)

[4 Test Process Description 5](#_Toc194339164)

[4.1 Test Planning Phase 5](#_Toc194339165)

[4.1.1 Entry/Exit Criteria for Each Phase 5](#_Toc194339166)

[4.1.2 Defects Severity and Priority Description 6](#_Toc194339167)

[4.2 Test Design Phase 6](#_Toc194339168)

[4.2.1 Test Cases Creation Rules 6](#_Toc194339169)

[4.2.2 Managing Test Cases in the Test Management Tool 7](#_Toc194339170)

[4.3 Test Execution Phase 7](#_Toc194339171)

[4.3.1 Defect Reports Raising and Management Rules 7](#_Toc194339172)

[4.3.2 Defect Lifecycle 7](#_Toc194339173)

[4.4 Test Reporting Phase 8](#_Toc194339174)

[4.4.1 Test Report Containment 8](#_Toc194339175)

[4.4.2 List of the Metrics to Track on a Project 8](#_Toc194339176)

[5 TEST STRATEGY DESCRIPTION 9](#_Toc194339177)

[5.1 Testing Methods 9](#_Toc194339178)

[5.1.1 Manual Testing (Primary Focus) 9](#_Toc194339179)

[5.1.2 Black-Box Testing 9](#_Toc194339180)

[5.2 Testing Types 9](#_Toc194339181)

[5.3 Testing Levels 9](#_Toc194339182)

[5.3.1 Unit Testing (Developers) 9](#_Toc194339183)

[5.3.2 Integration Testing 9](#_Toc194339184)

[5.3.3 System Testing 9](#_Toc194339185)

[5.3.4 User Acceptance Testing (UAT) 10](#_Toc194339186)

[5.4 Testing Pyramid 10](#_Toc194339187)

[6 risk management 10](#_Toc194339188)

[7 Resources 11](#_Toc194339189)

[7.1 Roles and Responsibilities 11](#_Toc194339190)

[7.2 Communication Plan 11](#_Toc194339191)

[7.3 Test Environment 11](#_Toc194339192)

[8 Test Schedule AND Test Deliverables 11](#_Toc194339193)

# INtroduction to Project and Project Specific Impact to Testing

OpenCart is an open-source e-commerce platform that allows businesses to create and manage online stores efficiently. It provides a user-friendly interface, a wide range of extensions, and supports multiple payment gateways and shipping methods.

Project phase: *Testing*

Technology stack: *PHP, MySQL, HTM, CSS, JS*

# Scope of Testing

## In Scope

|  |  |
| --- | --- |
| • Home Page  • Registration  • Login  • Logout  • Search  • Add to Cart  • Product Compare & Add to Compare  • Product Display  • Checkout         • Shopping Cart | • Wish List  • My Account Page  • Order History Page  • Transactions Page  • Downloads Page  • Contact Us  • Currencies |

## Out of Scope

1. Any features not listed under the **Scope** section
2. Third-party integrations, including external payment gateways
3. Automated testing
4. Performance testing
5. Database testing
6. Security testing
7. API testing
8. **Only manual testing will be conducted**

## Third-Party Systems

|  |  |  |
| --- | --- | --- |
| System/component/interface under test | Description | Responsible side |
| |  | | --- | | **Payment Gateways** |  |  | | --- | |  | | Integration with Visa for processing transactions. | |  | | --- | | Third-party providers |  |  | | --- | |  | |
| |  | | --- | | **Shipping Services** |  |  | | --- | |  | | |  | | --- | | Integration with FedEx, UPS, DHL, for real-time shipping rates. |  |  | | --- | |  | | |  | | --- | | Third-party providers |  |  | | --- | |  | |

# Quality & Acceptance Criteria & Critical Success Factors

Quality and Acceptance Criteria:

1. All core functions (registration, product management, cart, checkout) must work as expected.
2. No **critical** or **high-severity** defects remain unresolved.
3. Discount, tax, and pricing rules apply accurately.

Critical Success factors:

1. All functionalities in the IN SCOPE section are covered.
2. Detailed and varied test cases are created.
3. Realistic test data is used for all scenarios.
4. All found defects are accurately reported and tracked.
5. Key e-commerce workflows are thoroughly tested (shopping, checkout).

# Test Process Description

## Test Planning Phase

### Entry/Exit Criteria for Each Phase

**Entry Criteria:**  
Business and functional requirements are documented.  
Resources (team, tools, test environment) are identified.

**Exit Criteria:**  
Test Plan document is reviewed and approved.  
Risk assessment and mitigation plan are completed.  
Test schedule and milestones are established.

### Defects Severity and Priority Description

|  |  |  |  |
| --- | --- | --- | --- |
| Severity | Priority | Meaning | Examples |
| Blocker | P0 (Critical) | Behavior causes the business/consumer to stop working. Application under test crashes or inoperable | * the business users are unable to continue a successful business operation in the system unless the issue is resolved; * loss or partial loss of key functionality; * system crash; * massive performance degradation. |
| Critical | P0 (Critical) | Behavior causes the business/consumer to operate in a limited way. No workaround available. | * loss or partial loss of key functionality; * operational error; * data integrity; * some performance degradation. |
| Major | P1 (High) | Behavior allows business/consumer to continue working in a limited way with work around. | * partial loss of functionality of the software, but allows the user to continue proceeding normal business operations; * usability/UI issues causing confusion due to inconsistency or ambiguity. |
| Minor | P2 (Medium) | Behavior has no functional impact on business/consumer and can also be attributed to UI aesthetics. | * the business user decides that the issue around this functionality is not vital to his use of the system; * a minor feature that is not functional in one module but the same task is easily doable from another module. |
| Trivial | P3 (Low) | The defect does not affect functionality or data, it is merely an inconvenience. | * cosmetics (font face/font size/text alignment; * misspelling. |

## Test Design Phase

### Test Cases Creation Rules

Each test case must follow a structured format to ensure clarity and consistency. The standard test case structure includes:

Test Case Id (e.g. TC\_MAF\_001);

Test Case Title (e.g. My Account: Navigation using 'My Account' Drop-down Menu);

Test Steps that include:

Pre-requisites (e.g. Open the application (<https://demo.opencart.com)>);

Test Data (e.g. Not Applicable);

Test Steps (e.g. 1. Click on 'My Account' drop-down menu; 2. Select 'My Account' option);

Expected Result (ER) (e.g. 2. User should be navigated to the 'My Account' page);

Priority (P0-P3);

Result (Pass/Fail);

Comments;

### Managing Test Cases in the Test Management Tool

Google Sheets is used to manage test cases.

## Test Execution Phase

### Defect Reports Raising and Management Rules

Defect Reporting Guidelines:

1. Each defect should be documented clearly and logged in the **Defect Log Sheet**.
2. Provide detailed steps to reproduce, expected vs. actual results, and attach screenshots if applicable.
3. Default assignee: **QA Lead or Assigned Developer** based on the issue type.
4. All defects should be reviewed by QA team before assigning to developers

Defect Log Fields:

Each bug report must follow a structured format to ensure clarity and consistency. The standard bug report structure includes:

Bug ID (e.g. Bug\_001);

Description (e.g. No confirmation email sent upon successful registration);

Steps to reproduce (e.g.

**Pre-requisites:**- Open the Application (https://demo.opencart.com) in Chrome Browser on Mac  
**Test Data:**- Not Applicable  
**Steps:**1. Click on 'My Account' Drop menu   
2. Click on 'Register' option   
3. Fill in the mandatory fields (First Name, Last Name, E-Mail, Password, and accept Privacy Policy)   
4. Click on 'Continue' button   
);

Actual Result (e.g. User is successfully registered but does not receive a confirmation email);

Expected Result (ER) (e.g. User should receive a confirmation email upon successful registration);

Severity (e.g. Major);

Priority (e.g. P1);

Status (e.g. Open);

### Defect Lifecycle

**Defect Statuses:**

1️. **Open** – QA logs a new defect.  
2️. **In Progress** – Developer starts working on the fix.  
3️. **Fixed** – Developer resolves the issue and marks it as fixed.  
4️. **Ready for Retest** – QA verifies the fix.  
5️. **Reopened** – If the issue persists, QA reopens it.  
6️. **Closed** – If the fix works as expected, the defect is closed.

**Available Resolutions:**

* **Fixed** – Issue is resolved and ready for retesting.
* **Won’t Fix** – Not critical or accepted as a known limitation.
* **Duplicate** – Already reported in another defect.
* **Cannot Reproduce** – QA or Dev cannot replicate the issue.
* **Deferred** – Postponed for a future release due to lower priority.

## Test Reporting Phase

### Test Report Containment

1. Introduction
   1. Purpose
   2. Scope
   3. Definitions, Acronyms, and Abbreviations
   4. Overview
2. Test Results Summery
3. Test Cases Executed
4. Suggested Actions
5. Diagrams
   1. Test Execution Summery
   2. Defect Severity Distribution
   3. Defect Priority Breakdown
   4. Defect Status Overview

### List of the Metrics to Track on a Project

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Metric name | Formula | Responsible role | Frequency | Way to provide |
| |  | | --- | | **Test Case Execution Rate** | |  |  |  | | --- | |  | | |  | | --- | | (Executed Test Cases / Total Planned Test Cases) \* 100% |  |  | | --- | |  | | |  | | --- | | QA Team |  |  | | --- | |  | | |  | | --- | | Weekly & After Each Release |  |  | | --- | |  | | |  | | --- | | Bar Chart |  |  | | --- | |  | |
| |  | | --- | | **Defect Density** |  |  | | --- | |  | | |  | | --- | | Total Defects / Total Test Cases Executed |  |  | | --- | |  | | |  | | --- | | QA Lead |  |  | | --- | |  | | |  | | --- | | After Each Iteration |  |  | | --- | |  | | |  | | --- | | Line Chart |  |  | | --- | |  | |
| |  | | --- | | **Pass/Fail Rate** |  |  | | --- | |  | | |  | | --- | | (Passed Test Cases / Total Executed Test Cases) \* 100% |  |  | | --- | |  | | |  | | --- | | QA Team |  |  | | --- | |  | | |  | | --- | | Daily & Weekly |  |  | | --- | |  | | |  | | --- | | Pie Chart |  |  | | --- | |  | |
| |  | | --- | | **Retest Success Rate** |  |  | | --- | |  | | |  | | --- | | (Successfully Retested Defects / Total Retested Defects) \* 100% |  |  | | --- | |  | | |  | | --- | | QA Team |  |  | | --- | |  | | |  | | --- | | After Each Iteration |  |  | | --- | |  | | |  | | --- | | Trendline Chart |  |  | | --- | |  | |
| |  | | --- | | **Production Defect Leakage** |  |  | | --- | |  | | |  | | --- | | (Defects Found in Prod / Total Defects Found) \* 100% |  |  | | --- | |  | | |  | | --- | | QA Lead |  |  | | --- | |  | | |  | | --- | | After Each Release |  |  | | --- | |  | | Line Graph |

# **TEST STRATEGY DESCRIPTION**

This section outlines the approach for testing the OpenCart e-commerce application, including the testing methods, types, levels, and industry-standard models such as the Testing Pyramid.

## **Testing Methods**

The testing approach for OpenCart will follow two primary methods:

### ****Manual Testing (Primary Focus)****

• All test cases will be executed manually without automation tools.  
• Testers will follow structured test cases to validate expected behavior.  
• Exploratory testing will be used to identify issues not covered by test cases.

• ChatGPT 4o (accessed via chatgpt.com) will be used to assist with generating test case ideas, generating and refining test cases, writing detailed test cases, paraphrasing content, and helping draft parts of the test plans, bug reports, and test result documents for clarity and conciseness.

### ****Black-Box Testing****

• Focuses on system behavior without examining internal code.  
• Testers will input values and validate the system's response.

## **Testing Types**

Multiple testing types will be performed to ensure the stability and reliability of OpenCart:

• **Functional Testing** – Verifies that the system meets business and functional requirements (e.g., login, checkout, cart operations).  
• **UI/UX Testing** – Ensures proper layout, design consistency, and usability across different devices.  
• **Regression Testing** – Confirms that new changes do not break existing features.  
• **Compatibility Testing** – Ensures the platform works on Mac OS – Chrome

• **Smoke Testing** – A quick check to verify that core functionalities work before deep testing.  
• **Sanity Testing** – Performed after a bug fix to ensure that the reported defect is resolved.  
• **Exploratory Testing** – Performed by testers without predefined test cases to identify edge-case defects.

## **Testing Levels**

The OpenCart testing process follows a structured approach based on standard testing levels:

### ****Unit Testing (Developers)****

* Conducted by developers before code is pushed.
* Focuses on testing individual functions and modules.
* Ensures that basic logic and data handling work correctly.

### ****Integration Testing****

* Verifies interactions between different modules (e.g., Login → User Dashboard, Add to Cart → Checkout).
* Ensures smooth data flow between components.
* Identifies issues caused by module dependencies.

### ****System Testing****

* Full end-to-end testing of OpenCart functionality.
* Verifies business flows such as user registration, product browsing, checkout, and payment processing.
* Performed by the testing team before release.

### ****User Acceptance Testing (UAT)****

* Conducted by stakeholders to validate business requirements.
* Ensures the application meets real-world user expectations.
* The final step before deployment.

## **Testing Pyramid**

The Testing Pyramid is an industry best practice that emphasizes a balance between different testing levels:

UAT (Minimal tests to validate business needs)

────

System Testing (Ensuring end-to-end workflows)

─────────

Integration Testing (Verifying component interactions)

────────────

Unit Testing (Most extensive; ensures code stability)

# **risk management**

The following risks have been identified and the appropriate action identified to mitigate their impact on the project.  The impact (or severity) of the risk is based on how the project would be affected if the risk was triggered.  The trigger is what milestone or event would cause the risk to become an issue to be dealt with.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Risk** | **Impact** | **Trigger** | **Probability** | **Mitigation Plan** |
| 1 | Re Resources are not available | High | Tester unavailability due to unforeseen reasons | 70% | Plan backup resources and distribute workload efficiently |
| 2 | Changes in functionality | Medium | OpenCart updates or feature modifications | 50% | Export test data before upgrades and re-import after necessary modifications |
| 3 | Limited time for testing | Medium | Compressed timelines or unexpected delays | 50% | Increase resources dynamically based on workload and priorities |
| 4 | Third party plugin conflicts | Medium | |  | | --- | | Installed extensions break core functionality |  |  | | --- | |  | | 50% | |  | | --- | | Test in a staging environment before deploying new plugins |  |  | | --- | |  | |

# Resources

## Roles and Responsibilities

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Project Role | Name | Location | Responsibilities |
| 1 | Software Tester | Dilshodbek Isokjonov | Tashkent, Uzbekistan | Create Test Plan/Test Cases/Test Result Report, Executing Test Cases, Reporting Bugs |

## Communication Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Issue | Responsible person/people | Contact point | Communication plan |
|  | Inform the team about critical defect | All test team member | All test team members | Via Teams |

## Test Environment

Operation Systems and Browsers:

1. Mac OS – Chrome Browser

Environments:

1. Development
2. QA
3. PROD

Tools:

1. Chrome Browser
2. Google Sheets
3. MS Word

# Test Schedule AND Test Deliverables

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Task Name** | **Deliverable** | **Start** | **Finish** | **Effort** | **Comments** |
| Test Plan Creation | Test Plan | Mar 01 | Mar 03 | 2 days | MS Word doc |
| Test Case Creation | Test Cases | Mar 05 | Apr 05 | 30 days | Link to Google Sheets  ([The Link](https://docs.google.com/spreadsheets/d/1qDKptxdbh4MDNpaXHH-oo1pKbJB7Aa86xMWqehAEF_c/edit?usp=sharing)) |
| Test Execution & Bug reporting | Test Execution Results & Bug Reports | Apr 05 | Apr 20 | 15 days | Link to Google Sheets  ([The Link](https://docs.google.com/spreadsheets/d/1-zwXrM73f7p7g7akMim5vOnhu3GzxKfqGHKz1a-hG6A/edit?usp=sharing)) |
| Test Results Document Creation | Test Results | Apr 20 | May 01 | 11 days | MS Word doc |