**Test Strategy**

[Spree Ecommerce]

**Version:** 1.0

**Table of Contents**

1.INTRODUCTION

1.1. Purpose

1.2. Project Overview

2.TEST STRATEGY

2.1. Objectives

2.2. Test Methodology

2.3. Levels of Testing

3. TEST ENVIRONMENT

4 DEFECT MANAGEMENT

5. TOOLS/ FRAMEWORK

**1.INTRODUCTION**

**1.1 Purpose**

The purpose of the Testing Strategy document is to provide a reference at any time for resources working on tasks within the testing process of Spree Ecommerce project. This document is the source of guidance for the testing activities to be followed for the project. All members of the team should understand and follow the same testing strategy.

**1.2 Project Overview**

Spree Ecommerce is an open source ecommerce framework. Spree has been used by numerous companies from different domains (Fashion, beauty, Health etc..) to build products like marketplace, ecommerce sites quickly by leveraging the underlying solution, thus enabling them to release their products to market faster.

**2. TEST STRATEGY**

**2.1 Objective**

The objective is to support Spree Ecommerce to

* Improve overall quality of the solution through rigorous testing
* Regain customer confidence by improving speed of defect identification and resolution and time to market for new features
* Improve regression testing efficiently so that new features does not impact existing features.
* Build safety net in the application
* Enhance the solution for integration with new payment,shipping systems
* Build and test omni-channel capabilities

**2.2 Test Methodology**

The test methodology that would be followed for the Spree Ecommerce is Agile.

In Agile/Scrum, the entire project is divided into a set of user stories and then each story is worked upon in a sprint. A sprint will be typically 2 weeks in scope.

Each story must have an acceptance criteria and all acceptance criteria should be testable. These criteria are used as a standard to measure the success of each test. Test scenarios are created from each acceptance criteria.

**Done Criteria**

A Definition of Done drives the quality of work and is used to assess when a User Story has been completed.

Definition of done is achieved when:

* The User story has been completed.
* QA complete
* All tests (unit, functional, integration) should pass
* The story meets and satisfies the acceptance
* Regression around the story is complete
* Meets all cross-functional requirements
* The feature is eligible to be deployed in production

Example of a Done Criteria for Spree eCommerce:

Any User is successfully able to login and do a purchase and can see the purchased product in his/her Order Page.

**2.3 Types of Testing**

The types of testing that will be followed for the Spree Ecommerce application are:

* **Unit testing**, to ensure code is developed correctly. For the Spree Ecommerce application, unit tests should be written for both functional and UI classes. Unit tests should also test boundary values and edge cases. Unit tests increase the safety net of the application.
* **API/ Service Testing**, to ensure communication between Spree Ecommerce and the third party applications (payment, shipping systems) are working. If the underlying APIs are not ready we can mock them as well for testing the flows.

## **Automated Acceptance Test**s - It include Integration Tests and Service Tests and UI tests which aim to prove the software works at a functional level and that it meets user’s requirements and specifications.

For example, in the Spree Ecommerce, integration test and service tests would cover from selecting a link on home page till checkout and payment gateway page.

* **Regression Testing** - There will be smoke pack and full regression pack.

Smoke pack – Should be no more than 15 mins

This pack contains only high-level functionality to make sure the application is stable enough for further development or testing.

For example, for the Spree eCommerce website tests included in this pack could be:

Account Creation/Account Login

Product Search,

Add the product to the cart

Purchase Item

Full regression pack

This pack contains the full regression suite of tests and contains everything else which is not included in the smoke pack.

For the Spree eCommerce website, a new regression suite would be created because one did not exist.

## **Non-functional Testing** - Performance and Security testing will be performed.

## In the Spree eCommerce website, performance testing will be carried out through load testing, wherein QA’s would try to evaluate how the application will perform when the number of users as well as the number of transactions increase significantly. Some transactions tend to be more network intensive than others, for example: Product Search. Test such transactions would be tested across different browsers to ensure optimal transaction processing speed.

## Security testing will be performed on the Spree eCommerce website and also penetration testing will be performed to secure the application. For example, it would check integration with third-party payment gateways is secured.

* **Cross Browser/Cross Device Testing**

Testing needs to be done across different supported browsers and supported devices(Mobile devices, tablets etc…). Cross browser testing should cover the main platforms like Linux, Windows, Mac etc..

The Spree ecommerce applications will be checked for compatibility to work on all types of web browsers such as Google Chrome, Firefox, Internet Explorer, Safari, etc

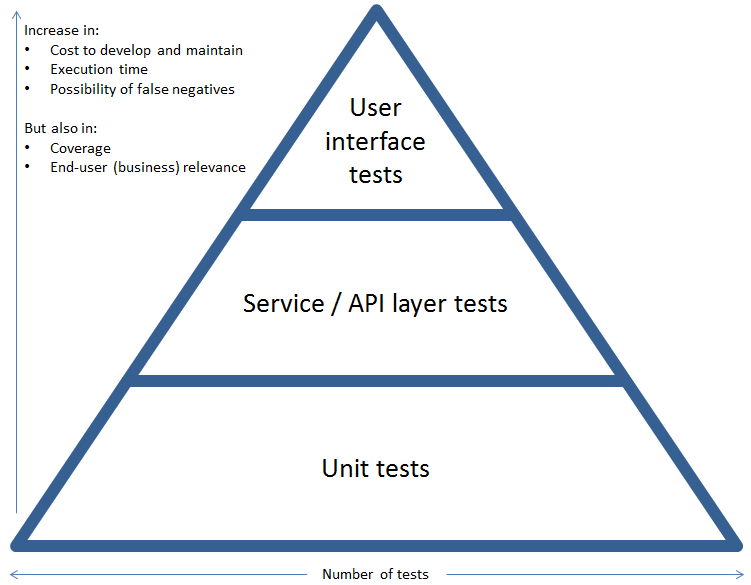
* **UAT and Exploratory** - Manual testing of the business flows to confirm the built product is what was expected and that it meets user expectations

Exploratory testing will focus on user scenarios and should find bugs that automation misses.

Plan would be done for any third party integration that needs to be facilitated for testing. Since the third party integration happens only at the later stage in the iteration, there should be stubs or mocks in place to leverage the integration capability for carrying out integration testing.

The mock/ stub should be designed well in advance before the Dev to QA happens , so that Integration testing can happen quickly and easily.Example of a Mock in our application is : Payment gateway which needs to be integrated for testing.

Each story in the Spree Ecommerce will follow the the test pyramid for testing an entire story. The approach to be followed here is Test Driven Development (TDD), where in the unit tests are developed before the development of code



The types of testing that will be carried out for the Spree Ecommerce application in the different environment:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Types of testing** | **Automated/ Manual** | **Environment** | **Executed By** | **Comments** |
| **Development-level testing** | | | | |
| **Unit, component, integration** | Automated | Development | CI | Continuous integration, faster feedback |
| **Deployment** | Automated | Development | CI | Continuous deployment |
| **Functional testing** | | | | |
| **Smoke** | Automated | All env | CI |  |
| **Sanity** | Automated | All env | CI |  |
| **Regression** | Both | QA/Regression env | Manual - QA, Regression team  Automation - CI |  |
| **Exploratory** | Manual | QA env | QA |  |
| **Compatibility testing** | | | | |
| **Browser** | Both | QA/Regression env | Manual - QA,  Automation - CI |  |
| **OS** | Both | QA/Regression env | Manual - QA,  Automation - CI |  |
| **Usability testing** | Manual | UAT env | QA/BA, Business users |  |
| **Performance testing** | | | | |
| **Response time performance** | Both | Dev env, QA env, Pre-prod env |  | Needs prod-like infra  Set up for generating reliable results |
| **Load Testing** | Both | Pre-prod env |  | Needs prod-like infra  Set up for generating reliable results |
| **Security Testing** | | | | |
| **Development security testing** | Both | Dev / QA env, | Manual - QA Dev  Automation - CI | Incorporated in daily development including story acceptance criteria |
| **Pen-testing** | Both | Pre-prod/ UAT env |  |  |
| **User Acceptance Testing** | Manual | UAT env | Business users |  |

**3. Environments**

Different environments that will be used for testing the Spree Ecommerce application:

|  |  |  |  |
| --- | --- | --- | --- |
| **Environment** | **Purpose** | **Data** | **Team Responsible** |
| **CI** | Certify build with execution of unit test, ,service test and functional test(smoke) | New Data set | Dev and Tester |
| **QA** | Exploratory testing and Regression testing (functional test) | Data subset | Dev and Tester |
| **Performance** | Performance testing and exploratory testing | Prod like data | Dev and Tester |
| **Pre-Prod** | Environment for business to validate the functionality | Prod like data | User and Tester |

**4. Defect Management**

The defect tracking tool, Jira will be used to manage the defects.

**Defects Creation**

QA when creating a Defect in Jira must include the following:

Subject

Defect Status

Severity/Priority – select from drop down list (see severity list below)

Defect Description to include:

o Description of Defect (what is the issue)

o The Requirement

o Steps to reproduce

o Expected results

o Actual Results

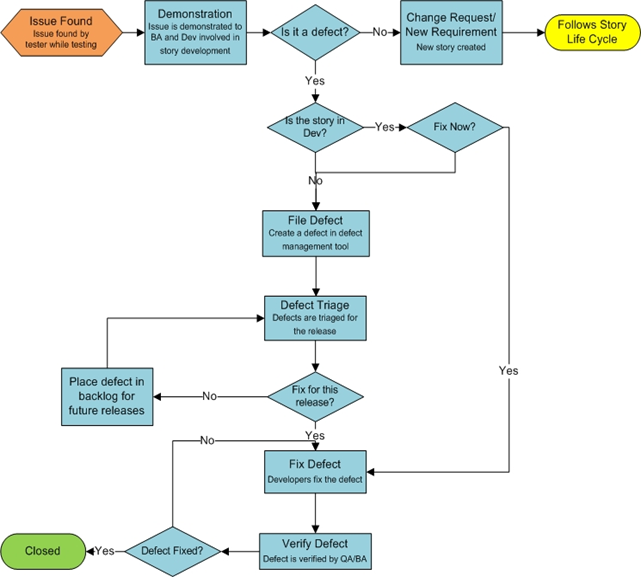
o Requirement #

o Design item #

o Test case #

· Screenshots (if applicable)

**Defect Life Cycle**



If defects are found during story testing, those defects would be raised in Jira.

The following section describes the different classifications that a Defect can have.

#### **Defect Severity**

|  |  |
| --- | --- |
| **Severity** | **Description** |
| **S1**  Critical | * Any system reported errors or incidents that stop processing or prohibits all tests from continuing. |
| **S2**  High / Major | * An error that prohibits the completion of an order or process as it was designed and configured. Requires correction before moving to the next test phase. |
| **S3**  Medium / Minor | * An order or process completes successfully with a workaround in place. |
| **S4**  Low / Trivial | * Other/Question. Any behavior identified that does not work the way it is expected or desired but does not stop successful completion of an test case or process and does not significantly impact business operations. |

#### **Defect Priority**

If more than one defect has the same severity, then the Priority will be used to determine which defect to fix first.

|  |  |
| --- | --- |
| **Priority** | **Description** |
| **P1** | Highest |
| **P2** | Medium High |
| **P3** | Medium Low |
| **P4** | Lowest |

5. **Tools/ Framework for Testing Spree ECommerce**

**Manual Testing:**

* Service / API tests - PostMan
* Functional UI tests - Web Browsers

**Automation Frameworks:**

* Selenium Webdriver- For UI Automation
* Rest Assured - for REST API testing
* TestNG - Test reporting tool
* Intellij - for developing the scripts.
* Maven - Build automation tool to facilitate CI/ CD pipeline
* Git - Version Control System to manage automation scripts
* Jenkins - Continuous Integration tool