Test Strategy

Spree E-commerce

Document Control

Document Detail

Title:	Test Strategy for Spree E-commerce website
Version:	10.1
Date:	06/05/19
Electronic File Name:	Spree-TestStrategy.doc
Electronic File Location:	https://github.com/sanajas
Author:	Sana Siddiqui
Contributors:	If anyone is contributing

Change Control

Issue Date	Version	Details	Author
06/05/19	10.1 V	Working Draft, not yet for review	Sana Siddiqui

Test Strategy Identifier

Spree E-Commerce Test Strategy-06-05-19

Introduction

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

Customers are happy that they are able to leverage the extensive capabilities provided by Spree E-commerce , however of late lot of customers are not happy with the quality of the solution in general.

Spree team is also planning to enhance their solution to enable them to enter new markets/regions. Entering new markets would also mean integration with local payment , shipping systems etc.

Scope and Overview

Scope and Overview:
The scope of testing for Spree E-commerce includes:

- Test the enhancements of applications to new regions
- Test the integration of existing applications like Payment and Shipping systems with the new applications.
- Test the Omni channel capability developed by Spree for performing multi level channeling mechanism. This would include functional testing, system testing, load testing, integration testing and performance testing and user experience testing.

The overall application development would be performed using the Agile methodology with Scrum adaptation. The stretch of each Sprint would be set to 2 weeks with the various ceremonies followed as per Scrum principles

Approach

The requirements will be split in to the below format to enable good development/test strategy:

Epic Features Story

User stories should be simple, concise and unambiguous.

A good user story should be:

Independent (of all others)

Negotiable (not a specific contract for features)

Valuable (or vertical)

Estimable (to a good approximation)

Small (so as to fit within an iteration)

Testable (in principle, even if there isn't a test for it yet)

User stories would be written using the below Specflow format:
As a [role] - As a customer

I want [feature] - I want the Login window with two factor authentication

So that [benefit] - so that I can access my account

Acceptance Criteria:
As part of each story, Acceptance Criteria needs to be defined explicitly and with no scope for ambiguity. Any queries/doubts around the story/requirement needs to be called out early in the refinement stage and ensure that either the PO or the business provides the necessary clarifications in the written format to enable proper testing.

Test Levels/Types:
The test team would be responsible for performing the following levels of testing for the applications:

- Unit Testing
- System Testing
- Integration Testing
- Performance Testing
- Regression Testing

Roles and Responsibilities:
The Project team would be responsible for clarifying the requirements prior to the project kick off from the Business/Product Owner to the extent of getting the initial Sprints clarity.
The role of Testers on the Agile Scrum Team is not limited to the Testers, but is extended to the Developers wherein they would be responsible for clarifying the requirements before development commences, and eventually would perform development unit testing on each of their modules in the Development Environment.

The Testers on the team would be responsible for engaging in 3 Amigos discussions with the developers and PO at every level of requirement refinement.
The Testers would perform manual and automated testing on every MVP (Minimal Viable Product) in the Test Environment after the code is developed and unit tested in Dev Box and would also be responsible for providing the sign-off to each of the requirements/code modules being showcased/demonstrated to the business.

Testing Tools:
The following test tools would be used for performing the testing on the applications in development:

- Selenium with Java for UI testing
- RestAssured for API testing with Java
- Jeera for Test Management and Defect tracking
- Perfecto for remote testing using various mobile devices

CI/CD

The Build Manager will ensure that once testing begins no changes or modifications are made to the code used to create the build of the product under test. The Build Manager will inform the Test Team against which version testing will begin and confirm the location within the build is to be taken from.

If changes or modifications are necessary through bug resolution or for any other reason the Build Manager will inform the Test Team prior to the changes being made

Measures and Metrics

At the Initiation Phase of the project the Test Team will publish a set of measures and metrics related to the test activities of their Planning & Analysis and Execution phases. The Test Plan also defines the milestone dates for key deliverables such as the Test Plan and these are metrics captured for ongoing statistical process analysis across successive projects.

‘Pass/Fail’ Criteria

Each Test Item will be assigned a Pass or Fail state dependant on two criteria:
Total number and severity of Bugs in an Open & Unresolved state within Jeera/Bug Tracker.
The level of successfully executed test requirements.

The combination of both criteria will be used to recognise the Test Item can be declared Test Complete. However as this is a minimum level of quality that is believed achievable it’s recommended that where project timescales allow further testing and development should be conducted to raise the overall quality level.

Table of Issue Severity

Severity	Definition	Maximum Allowable
S1	Crash/Legal – System crash, data loss, no workaround, legal, Ship Killer	0
S2	Major – Operational error, wrong result	<Set by Business>
S3	Minor – Minor problems	<Set by Business>
S4	Incidental – Cosmetic problems	<Set by Business>
S5	N/A – Not Applicable; used for feature requests and Development Tasks	

The total MAXIMUM number of issues recorded in Jeera / Bug Tracker that can remain in an Open & Unresolved state for the Test Item and be acceptable for release.

Table of Test Scenario Priority

Test Scenario	Definition	Minimum Pass Rate
P1 – Critical	Essential to the Product	100%
P2 – Important	Necessary to the Product	<Set by Business
P3 – Desirable	Preferred, but not essential to the Product	<Set by Business>

The MINIMUM set of Test Scenarios that must pass before the Test Item can be considered for release.

Test Deliverables

The following artefacts will be produced during the testing phase:

Test Plan
Used to prescribe the scope, approach, resources, and schedule of the testing activities. To identify the items being tested, the features to be tested, the testing tasks to be performed, the personnel responsible for each task, and the risks associated with this plan.

Test Schedule
Which describes the tasks, time, sequence, duration and assigned staff.

Test Breakdown
Which includes the Test Scenarios, their priority and related number of Test Cases along with the defined estimates for time to write and execute the Test Cases.

Test Cases
Detail the pre-conditions, test steps and expected and actual outcome of the tests. There will be positive and negative test cases.

Defect Metrics
Defect Data Generated across various sprints along with prioritisation and status

Status Report
Regular management updates as agreed with the client.

Testing Tasks

The Testing Tasks that the Test Team will deliver cover the following scope:

Fully In Scope: Functional and Regression Testing

Partially in Scope: Cross Browser Compatibility, Integration in the Large.

Out of Scope: Performance testing, Automated Regression, all forms of Non-Functional, Accessibility Compliance Testing, Security Testing, User Documentation Review.

Environmental and Infrastructure Needs

The following detail the environmental and infrastructure needs required for the testing of lastminute.com Test Items and execution of Regression Testing.

Hardware.
Integration Environment:
Test-A: http://.....
Test-B: http://....
Pre-live Staging:

Software

<Name of Bug Tracking Tool>: http://.Jeera..
<Name of Test Case Management Tool>: http://trello
<Name of Automation Tool>: http://selenium

Infrastructure
Network connections are available on all Test Systems as required.

Test Repository
http://github.

I				

Approvals
The following people are required to approve the Test Strategy

Approval By	**Approval**
PO	
Project Manager	