

ThoughtWorks®

Test Strategy

Spree Ecommerce

Table of Contents

1.	INTRODUCTION	3
1.1.	PURPOSE	3
2.	TEST ITEMS	3
3.	FEATURES TO BE TESTED	3
4.	FEATURES NOT TO BE TESTED	3
5.	APPROACH	4
5.1.	ANALYSIS & PLANNING PHASE ENTRY CRITERIA	4
5.2.	ANALYSIS & PLANNING PHASE EXIT CRITERIA	4
5.3.	TEST PHASE ENTRY CRITERIA	4
5.4.	TEST PHASE EXIT CRITERIA	4
5.5.	CHANGE MANAGEMENT	5
5.6.	NOTIFICATION / ESCALATION PROCEDURES	5
5.7.	MEASURES AND METRICS	5
6.	'PASS/FAIL' CRITERIA	6
7.	SUSPENSION CRITERIA & RESUMPTION REQUIREMENTS	7
8.	TEST DELIVERABLES	7
9.	TESTING TASKS	8
10.	ENVIRONMENTAL AND INFRASTRUCTURE NEEDS	8
11.	RESPONSIBILITY MATRIX	8
12.	STAFFING AND TRAINING NEEDS	9
13.	SCHEDULES AND RESOURCE PLANS	9
14.	RISKS AND CONTINGENCIES	9
15.	APPROVALS	10

1. Introduction

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. Purpose and Objectives

- This document provides you the Test Plan and strategy that QA Engineers will follow for the Spree Ecommerce project. This document is intended to be used by the QA engineers of ThoughtWorks.
- Provide a framework for testing any product developed using the Spree Ecommerce framework
- Document the different types of testing that needs to be done to deliver a fully functional high quality product.
- Provide recommendations on usage of appropriate tools, test approach, automation framework, defect management process.

3. Features to be tested

The Test Engineer will use the Test Breakdown worksheet to record all of the features to be tested for each of the Test Items in scope.

The Test Breakdowns will include details of the Test Scenarios from which the Test Cases will be derived.

Following are the modules which need to be verified for the accuracy of the implementation. These requirements would be tested as part of testing activities release-wise:

Req. ID	Requirement Title
1	Home Page
2	Login Page
3	Product Listing Page
4	Product Details Page
5	Add To Cart Page
6	Check Out Page
8	Shipping System

4. Features not to be tested

- Payment Gateway as it the third party vendor

5. Approach

The first step is to define the testing strategy. It should describe how and when the testing will be conducted, who will do the testing, the type of testing being conducted, features being tested, environment(s) where the testing takes place, what testing tools are used, and how are defects tracked and managed. The testing strategy should be prepared by the agile core team. In an agile project:

- Feature testing occurs throughout each sprint with the final user acceptance testing (UAT) conducted in the testing sprint before going live.
- During the release planning meeting, the team should capture acceptance criteria and immediately add them as logical test cases linked to the product backlog item.
- In each sprint planning meeting, a sprint test plan should be created and reviewed.
- As the testing is done in each sprint, the results should be tracked along with which features have been successfully tested and which ones have defects. These results should be passed back to the CA consultant who can fix the defects.
- Care should be taken to distinguish defects and changes in requirements. If the team requires a change in requirement, this should be added to the product backlog and prioritized.
- Testing for cross browser and cross device testing (QA Sprint needs to be planned and thought out thoroughly on what needs to be covered in these tests)
- When an App is expanded to a particular region, we need to be clear on what user stories need to be picked up and tested upon to release it faster and defect free.
- Cross border trading can be thought of as a user Story and specific Acceptance criteria can be fixed for those stories.

6. Test Planning

The steps involved in planning and executing testing are:

- In each sprint, product owner shares the customer voice to the scrum master and in turn it is being shared with scrum team.
- Everybody should have the same understanding of what the story is about.
- Product owner should create the prioritized product backlog and define the acceptance criteria.
- Scrum team should have a good understanding of the technical details that are involved in delivering the story.

- In the sprint planning meetings, the estimates given for a story should include the testing effort as well and not just coding effort.

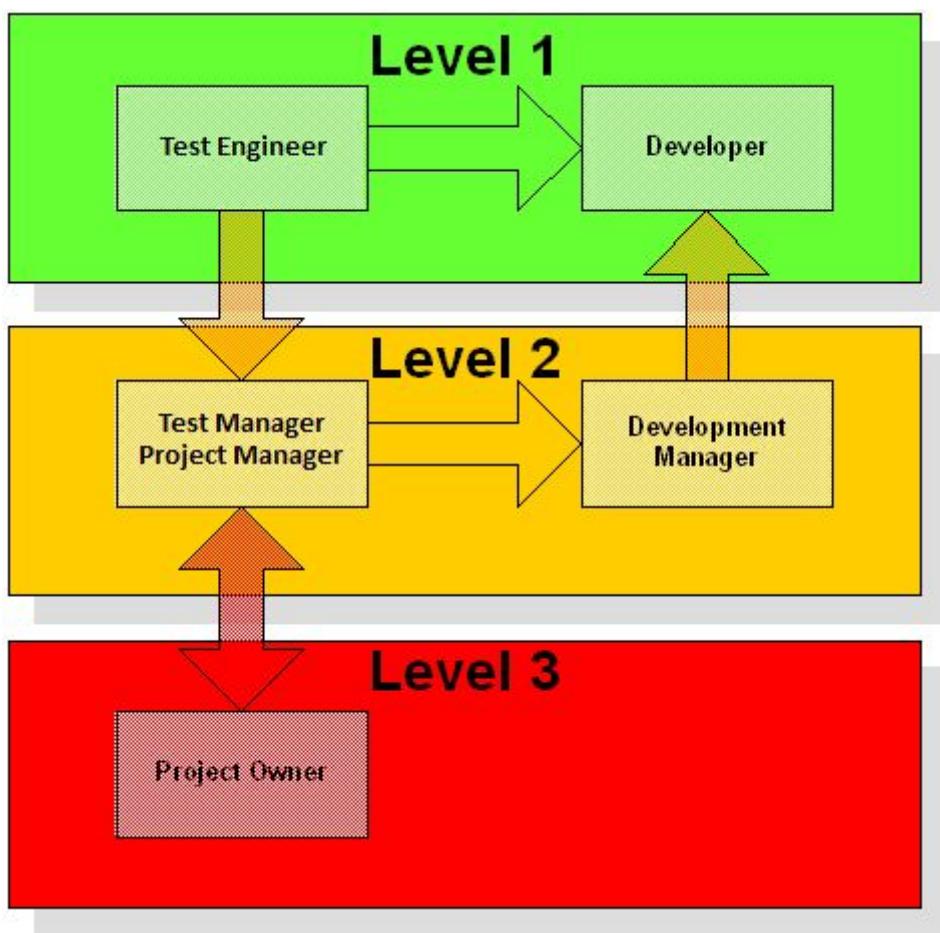
6.1. Change Management

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 [VSS/Progress/Perforce/Subversion] 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.

6.2. Notification / Escalation Procedures

The following diagram shows the notification and escalation paths to be followed for the duration of the project Test Phase.



6.3. 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.

Test Preparation

- Number of Test Scenarios v. Number of Test Cases
- Number of Test Cases Planned v. Ready for Execution
- Total time spent on Preparation v. Planned time

Test Execution and Progress

- Number of Tests Cases Executed v. Test Cases Planned
- Number of Test Cases Passed, Failed and Blocked
- Total Number of Test Cases Passed by Test Item / Test Requirements
- Total Time Spent on Execution vs Planned Time

Bug Analysis

- Total Number of Bugs Raised and Closed per Test Run
- Total Number of Bugs Closed v. Total Number of Bugs Re-Opened
- Bug Distribution Totals by Severity per Test Run
- Bug Distribution Totals by Test Item by Severity per Test Run

7. Testing Types

In order carry out end to end application testing following testing is been carried out

- Unit Testing
- API Testing
- Integration Testing
- System Testing (Smoke, Sanity)
- Database Testing
- Functional Testing
- UI Testing
- Regression Testing
- Cross Browser testing
- Usability Testing
- Localization Testing
- User Acceptance Testing

8. 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 Bugzilla/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 PM>
S3	Minor – Minor problems	<Set by PM>
S4	Incidental – Cosmetic problems	<Set by PM>

S5	N/A – Not Applicable; used for feature requests and Development Tasks	Reference Only
----	---	----------------

The total MAXIMUM number of issues recorded in Bugzilla / 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 PM>
P3 – Desirable	Preferred, but not essential to the Product	<Set by PM>

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

Unforeseen issues arising during the Test Phase may impact the agreed ‘Pass/Fail’ Criteria for the Test Item. Issues can be managed through review with the Test Team and the project authorities.

9. 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.

- **Periodic progress and metric update reports**

- **Bug Reporting**

- **Test Summary Reports**

- **Automated Scripts**

10. Tools and Framework used for this project

Manual Testing:

- Service / API tests - PostMan, SOAP UI
- Functional UI tests - Web Browsers (Chrome, FireFox, Safari)

Automation Frameworks:

- Appium - For Mobile devices Automation
- Selenium - For UI Automation
- Rest Assured - for REST API testing

TestNG - Test Automation tool and reporting tool

Maven - Build Management Tool

Git - Version Control System to manage automation scripts and enable central storage of all tests

11. Testing Execution using Test pyramid

As part of the Test Automation Strategy, we need to ensure to minimize the number of automated tests that are run at GUI layer.

Whilst running automated tests through the GUI provides good and meaningful tests in terms of simulating a user's interaction with the application, it is prone to many issues as listed below:

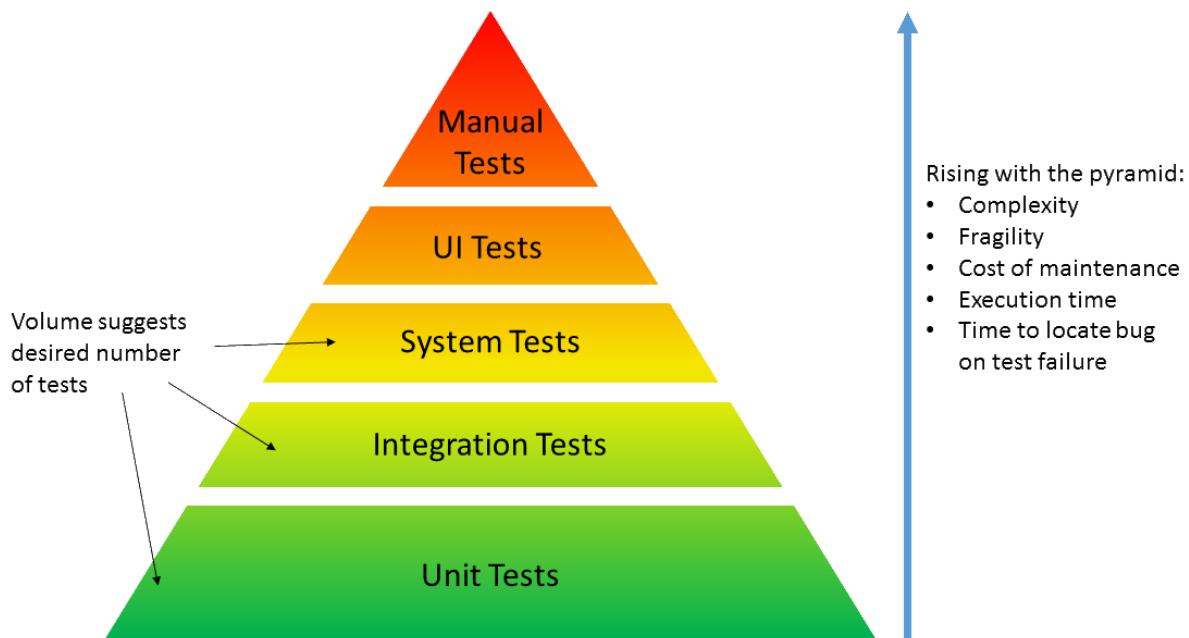
Brittle – Because the tests rely on the HTML locators to identify web elements to interact with, as soon as an id is changed the tests fail, therefore they bear a lot of maintainability costs.

Limited Testing – The GUI could confine the tester's ability to fully verify a feature as the GUI may not contain all the details from the web response to allow verification.

Slow – Because tests are executed through the GUI, the page load times can substantially increase the overall testing time and as such the feedback to the developers is relatively slow.

Least ROI – Because of the above-mentioned issues, the GUI automated tests provide the least ROI.

The Browser Automation Tests will be kept to a minimum and will be used to simulate a user's behavior incorporating common user flows and end-to-end scenarios where the system as a whole is exercised.



12. 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://...
- <Name of Test Case Management Tool>: http:///
- <Name of Automation Tool>: http://

Infrastructure

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

Test Repository

- http://...

13. Responsibility Matrix

The table below outlines the main responsibilities in brief for test activities:

Activity	Product Manager	Development Manager	Test Manager	Test Engineer
Provision of Technical Documents	X	X		

Test Planning and Estimation			X	X
Review and Sign off Test Plan	X	X	X	
Testing Documentation			X	X
Test Preparation and Execution				X
Test Environment Set-up				X
Change Control of Test Environments			X	X
Provision of Unit Tested Test Items		X		
Bug fixes and return to the Test Team for re-test		X		
Product Change Control	X	X	X	
Ongoing Test Reporting			X	X
Test Summary Reporting			X	

14. Staffing and Training Needs

Staffing.

Staffing levels for the test activities will be:

- 1 x Test Manager for the duration of test planning at 50% effort against plan.
- The required number of Test Engineers for the duration of test execution at 100% effort against plan.

Training.

For each project the training needs will be assessed and defined in the Test Plan.

15. Schedules and Resource Plans

Team Plan.

The Test Team will maintain a Team Plan which records individual assignment to testing tasks against assignable days. This will also record time planned and delivered against the tasks which will be used to update relevant Project Schedules and be used in periodic reporting.

Test Schedule.

The Test Schedule for the Release will be located within <Spree Ecommerce> at: <http://>

16. Risks and Contingencies

The agile core team should develop a comprehensive list of all potential risks. These types of risks may include:

- Resource Availability
- Financial
- Technical
- Scope

- Business Process
- Training/User Adoption
- Communication

These risks should be prioritized based on likelihood and potential impact. This information can be used by the agile project team to focus on developing plans to mitigate these risks and to develop contingencies. Just like on any other project, if a risk management methodology is readily available, it should be used.