

**X&X**

**2017**

**Test Strategy Document**

|  |  |
| --- | --- |
| Title | X&X 2017 Test Strategy |
| Author | XXX |
| Version | 1.0 |
| Date | 20th February 2017 |

Contents

[1. Overview and Scope 3](#_Toc474466570)

[1.1 In Scope of XXXXXX Test 3](#_Toc474466571)

[1.2 Out of Scope of XXXXXX Testing 3](#_Toc474466572)

[2. Test Approach 4](#_Toc474466573)

[2.1 Overview 4](#_Toc474466574)

[2.1.1 Test Process 4](#_Toc474466575)

[2.2 Manual Testing 4](#_Toc474466576)

[2.3 Cross Browser and Device Testing 5](#_Toc474466577)

[2.4 Web Vulnerability Testing 6](#_Toc474466578)

[2.5 Email Testing 6](#_Toc474466579)

[2.6 Testing the business tooling 6](#_Toc474466580)

[2.7 Validating 3](#_Toc474466581)[rd](#_Toc474466581) [Party APIs 6](#_Toc474466581)

[2.8 File Based Interface Testing 7](#_Toc474466582)

[3. Defect Management 7](#_Toc474466583)

[3.1.1. Defect Management – In System Testing 7](#_Toc474466584)

[3.1.2. System Testing Defect Flow 7](#_Toc474466585)

[3.1.3. UAT Defect Flow 8](#_Toc474466586)

[3.1.4. Defect Priority Definitions 9](#_Toc474466587)

[3.1.5. Defect Template 11](#_Toc474466588)

[4. Roles and Responsibilities 11](#_Toc474466589)

[5. Test Environment 12](#_Toc474466590)

[5.1.1. SYSTEST 12](#_Toc474466591)

[5.1.2. UAT 12](#_Toc474466592)

[5.1.3. SIT 12](#_Toc474466593)

[6. release Environment 12](#_Toc474466594)

[6.1 Release process to UAT 12](#_Toc474466595)

[6.2 Release notes 13](#_Toc474466596)

[7. Entry and Exit Criteria 14](#_Toc474466597)

[7.1 Entry Criteria 14](#_Toc474466598)

[7.2 Exit Criteria 14](#_Toc474466599)

[8. Testing Tools 14](#_Toc474466600)

[8.1 Defect Management 15](#_Toc474466601)

[9. Tracking and reporting 15](#_Toc474466602)

Document History

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision Date** | **Version** | **Summary of Changes** | **Revised by** |
| 03/01/2017 | 0.1 | Initial Draft | Yan Yan |
| 12/01/2017 | 0.2 | Revision after Nick’s review | Yan Yan |
| 17/01/2017 | 0.3 | Update testing tools section after discussion with Terry | Yan Yan |
| 07/02/2017 | 0.4 | Update for environment and security testing section | Yan Yan |
| 10/02/2017 | 0.5 | A few updates | Yan Yan |
| 13/02/2017 | 0.6 | A few updates | Yan Yan |
| 20/02/2017 | 1.0 | Final internal version | Yan Yan |

Document Approval

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Title** | **Approved (y/n)** | **Date of Approval** | **Version Approved** |
| Philip Seale | XXXXXX Project Manager | **y** | **20/02/2017** | **1.0** |
| Terry Rowlinson | XXXXXX Technical Architect | **y** | **20/02/2017** | **1.0** |
| Charlotte Vaughan | XXXXXX Business Analyst | **y** | **20/02/2017** | **1.0** |
| Nick Gee | XXXXXX Head of Testing | **y** | **20/02/2017** | **1.0** |
| Stuart Clements | X&X Programme Manager |  |  |  |
| Ben Rees | X&X Head of Ecommerce |  |  |  |
| Russell Anthony | X&X Head of Digital Change |  |  |  |
| Deena Patel | X&X UAT Test Analyst |  |  |  |
| Sanjay Pappachan | X&X UAT Test Analyst |  |  |  |

Document Distribution

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Title** | **Date of distribution** | **Version Distributed** |
| Philip Seale | XXXXXX Project Manager |  |  |
| Terry Rowlinson | XXXXXX Technical Architect |  |  |
| Charlotte Vaughan | XXXXXX Business Analyst |  |  |
| Sajeedkhan Patagaonkar | XXXXXX Backend developer |  |  |
| Brian McHugh | Front End Developer |  |  |
| Stuart Clements | X&X Programme Manager |  |  |
| Ben Rees | X&X Head of Ecommerce |  |  |
| Russell Anthony | X&X Head of Digital Change |  |  |
| Deena Patel | X&X UAT Test Analyst |  |  |
| Sanjay Pappachan | X&X UAT Test Analyst |  |  |
| Adetola Coker | X&X Business Analyst |  |  |

# Overview and Scope

The purpose of this document is to:

* Define overall testing strategy to be adopted by XXXXXX.
* Define the high level testing approach to be taken by XXXXXX.

The X&X project is to be run using Agile Methods. Fundamental principles are that:

* Testing will run alongside development,
* Functional testing and regression testing will be manual in nature,
* Exploratory testing will be used where possible allowing testers to incorporate observed behaviour into their test design,
* JIRA and the XRAY test plugin will be used to monitor and track progress of testing in real time.

## In Scope of XXXXXX Test

* Creation and maintenance of all test cases (will be held in XRAY),
* Execution of all planned test cases created by the XXXXXX testing team,
* Compatibility Testing to verify each wireframe against the delivered code to ensure the look and feel of each page is as per design for phones, tablets and desktop (refer to section 2.3),
* Regression Testing across the sites listed below:
  + UK site including claims
  + Germany site
  + Sky Protect
  + Play Station
  + Italy site
  + Any other upcoming sites, e.g. Australia
* Integration Testing with 3rd party systems, e.g. Worldpay, X&X web services, Skyline APIs.
* Web Vulnerability Testing, using AppCheck and manual checks covering the OWASP top 10 on all new functionalities.

## Out of Scope of XXXXXX Testing

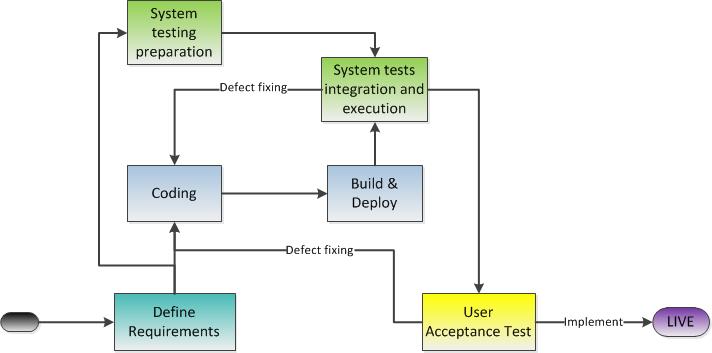
* Content testing, all the content related pages, eSpots, etc.
* Translation Correctness,
* Performance Testing,
* Penetration Testing,
* Automation Testing,
* 3rd Party application testing,
* User Acceptance Testing,
* Any other phase of testing not detailed within the In Scope section above.

# Test Approach

## Overview

The process of testing to be followed by the Agile teams is listed below.

## 2.1.1 Test Process



## Manual Testing

A fundamental principle is that Functional Testing will be carried out using 2 manual testing approaches:

* Scripted Testing
* Exploratory Testing

Scripted Testing

Complex features and those areas requiring client review of testing will be scripted. i.e. a set of detailed instructions will be written and expected results provided to ensure all Test Conditions are exercised.

Exploratory Testing

This approach ensures testers are involved in minimum planning and maximum test execution. The planning involves the creation of a test charter, a short declaration of the scope of a short (1 to 2 hour) time-boxed test effort outlining the objectives and possible approaches to be used. Test scripts are not required.

## Cross Browser and Device Testing

We will execute System Tests across a selection of browsers and devices. This will allow us to visually and functionally test all layouts of the responsive site, and reduce the occurrence of browser / platform specific defects.

The browsers and devices should be selected based on current live usage statistics. Typically testing will focus on the most popular devices and browsers, but ensure a cross section of device sizes and operating systems are included.

Our approach to this would be:

Desktop browser testing:

* The aim is to test only the desktop versions of the site visually and functionally.
* Time-boxed exploratory testing on each supported browser (we do not execute all scripted tests in all browsers).
* To be completed on the work delivered for each release.

Browsers list on desktop as below:

|  |  |
| --- | --- |
| **Browser** | **Version** |
| **IE** | 11 |
| **Microsoft** | Edge |
| **Firefox** | Latest version |
| **Chrome** | Latest version |
| **Safari** | Latest version |

Device testing:

* The aim is to test the mobile and tablet versions of the page ensuring that differing break points are correctly rendered and the screen fluidity and functions remain consistent.
* Time-boxed exploratory testing of X&X Italy site on each supported device
* To cover the main page layouts, functions and flows through the site
* This includes switching between portrait and landscape orientation on all pages
* To be completed on the work delivered for each release.

Browsers list on mobile/tablet as below:

|  |  |
| --- | --- |
| **Browser** | **Device** |
| **Safari** | iPhone X |
| **Safari** | iPhone 6s Plus |
| **Safari** | iPad Pro |
| **Native browser** | Samsung Galaxy S6 |
| **Chrome** | Samsung Galaxy S6 |
| **Native browser** | Samsung Tab 4 |
| **Chrome** | Samsung Tab 4 |
| **Native browser** | Nokia Lumia 930 |

## Web Vulnerability Testing

XXXXXX will test new functionalities for Web Vulnerabilities. This uses 2 techniques:

* Automated Vulnerability Scanning using AppCheck
* Manual security tests following XXXXXXs standard site checklist

Automated Vulnerability Scanning

XXXXXX has uses AppCheck NG to undertake vulnerability scans of websites. The tool uses a Spidering tool and its own scripting language to run a set of automated tests that probe for security vulnerabilities. The tools tests are based on the OWASP top 10 web vulnerabilities.

Manual security tests

In addition to the automated scanning the security testing team runs a set of tests defined and documented in the XXXXXX Manual Security Testing Approach document. These tests are run in conjunction with BurpSuite (a security testing toolkit and framework).

## Email Testing

On a high level, testing the emails, either transactional or non-transactional, will comprise of one of more of the aspects listed below:

* Validating the email templates for existence of all necessary fields, variables and configuration.
* Emails that are generated and sent by WCS should be validated for sender and recipient details, subject and body details, layout, content, calls to action, context/ dynamically populated variables, product information, product images, promotional information / codes as per requirements.
* If any email interfaces are built for triggering emails, recommendations etc., these should be specifically tested.
* If rendering of emails is to be tested, sprint teams should define the devices and browsers to be covered. If several combinations were to be tested in a short timeframe, an email-compatibility tool should be explored.
* Check emails are secure i.e. no sensitive information is stored in email html

## Testing the business tooling

On a high level, testing the WCS tooling to support business processes will comprise of:

* Testing customisations to existing WCS tooling. Primarily including, but not limited to, management centre.
* Testing any customised management tools prepared for X&X.

## Validating 3rd Party APIs

* These are internal X&X or 3rd Party back-end services such as plan services and skyline APIs.
* Where the 3rd party APIs is not available in the test environment, or cannot be integrated in the timelines required by the component team, services to/from this system will be mocked.
* Where a 3rd party system is within XXXXXX deliverables we will endeavour to test services to/from this system with WCS in an integrated fashion.
* Message formats and content will be tested using Soap UI or Postman or reading from logs.
* The tests will be created to exercise the APIs in a user story context to cover business logic.
* E.g. We can string services by first creating some data using a POST request, and then using another service to retrieve the same data with a GET request. For instance, doing this, you could simulate registering a X&X customer, adding an address to the customer’s account & checking that the same address appears during checkout.

## File Based Interface Testing

This testing covers those interfaces where send or receipt functionality is invoked for .txt, .csv, .xml etc. type files that may be transferred to and or from WCS. This testing will typically cover:

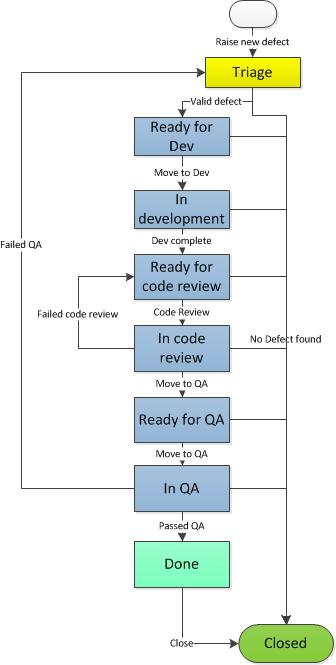
* Check the file format and content.
* Validate that the dataload utilities ingest/ load data in the WCS Database tables correctly. Inspect the data loaded.
* There may be pre-processing required before the data is loaded into WCS. Where this is the case, testing will validate that data processed correctly.
* Check validations (such as header validations, incomplete data sets etc.) and error handling,
* Use scenario based testing to simulate real life data situations. It is important to maintain a balance between acceptance criterion and test conditions. e.g. It may not be a valid business process to delete categories, only un-publish them. Testing category deletes may be considered invalid in this situation.
* Simulate test data that exercises insert/ update/ delete operations individually and together where appropriate. e.g. An allowed attribute value may be deleted and a new value for the same attribute created at the same time. Striking a balance to achieve real business like scenarios is key here.

# Defect Management

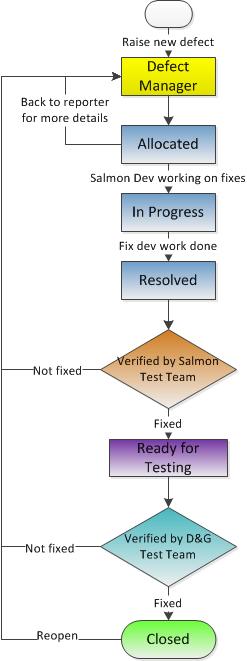
## Defect Management – In System Testing

During system development and testing, any issues identified will be logged and tracked through JIRA.

## System Testing Defect Flow



## UAT Defect Flow



## Defect Priority Definitions

The following definitions will be used by the XXXXXX System test team when raising a defect to set the defect priority.

|  |  |  |
| --- | --- | --- |
| **Severity** | **Description** | **Examples** |
| 1-Critical | Defect renders all or a portion of the system under test inoperable, meaning testing cannot continue. A Severity 1 defect could also be raised through the cumulative impact of a number of Severity 2 defects. | * SOLR index failing so I cannot surface any product information through browse or search services. The deployment has failed and the test environment is unavailable * Login is returning an error so I can’t test any scenarios related to known customers. |
| 2-High | An area of functionality does not work and is preventing test scripts being run or completed. There are no workarounds that can mitigate the impact of the defect. | * Certain search terms are throwing an error when using the search services * Data load is failing to load category description * You cannot register during checkout scenario |
| 3-Medium | A defect is identified that does not prevent the completion of an end-to-end test, but does cause the failure of a specific test step within that end-to-end test process; and in respect of the defect, there is a workaround agreed with the product owner which allows that area of functionality to be used within the relevant business processes. | * A validation of one of the json fields is incorrect. We can use the coded validation, but integration would fail * You can complete all the steps in the script but some may fail * The issue is only reproducible for a certain configuration of a product |
| 4-Low | A defect that relates to an extreme test condition and has minimal or no business impact. | * Layout inconsistencies * Defects that don’t materially impact the user experience – some compatibility issues |

## Defect Template

When raising a defect there are a number of key pieces of data that need to be captured. The template below provides a template for guidance. This includes the typical information that would be expected to be included in a defect.

|  |  |
| --- | --- |
| Field | Value |
| Summary | [Site] - [Functional Area] – [Summary] |
| Description | **\*Setup\*** - List the activities / data required to complete the test - set the scene  **\*Steps\*** - Only list the steps relevant to the failure - i.e. the bare minimum that causes the failure each time.  **\*Expected Result\*** - Clearly state what you expected to happen  **\*Actual Result\*** - Clearly state what is failing - Attach a relevant screen of the failure  **\*Additional Info\***  - Log in details  - Plan number(s)  - Design templates & Build versions  - Browser type and device details |

# Roles and Responsibilities

|  |  |
| --- | --- |
| **Testing Role** | **Key Responsibilities** |
| XXXXXX Test Lead | * Representative of the XXXXXX Testing Team * Liaison for interdepartmental interactions (where necessary) * Development of testing approach (This document) * 3rd Party interaction (where applicable) * Test-tool selection and introduction * Cohesive integration of test and development activities * Test-process definition, training and continual improvement * Test-program oversight - support delivery teams * Provide Technical expertise required for Testing * Support for customer interface, test-tool introduction, test-environment configuration * Test-script development * Review Testing Deliverables from other Testers * Share technical expertise and standards with other team members * Actively participate in review of component requirements with the product owner * Communicate and collaborate within the team effectively and with agility * Ensuring that test artefacts for the team are complete * Identify & report risks, mitigating them where possible |
| XXXXXX Test Analyst | * Design/ develop test cases based upon requirements/ user stories * Design, develop and execute reusable and maintainable automated scripts as per adopted framework * Follow test-design standards and take advice from Test Lead * Execute tests and publish reports as necessary * Actively participate in review of component requirements with the product owner * Communicate and collaborate within the team effectively and with agility * Ensure that test artefacts are complete * Identify risks, mitigating them where possible |
| UAT Test Analyst | * Coordinate with data team to provide enough data for system testing * Triage UAT defects and assign them accordingly * Responsible for UAT scripting, execution and reporting. |

# Test Environment

The project team will define a delivery pipeline appropriate for the team’s delivery ambition.

## SYSTEST

The sprint team will use SYSTEST for all functional testing:

* The test environments are hosted by XXXXXX Technical Service.
* A single server test environment is sized to run a single instance of WCS.
* The environment manifest will define which components get deployed into an environment.
* 3rd party Integration testing will be performed on SYSTEST environment as well.

## UAT

X&X test team will use UAT for user acceptance testing:

* The test environments are hosted by Rackspace.
* The environment manifest will define which components get deployed into an environment.
* 3rd party Integration testing will be performed on UAT environment as well.

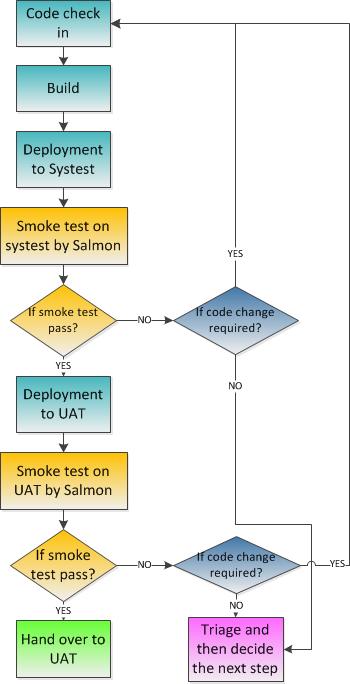
## SIT

XXXXXX TS team will use SIT environment for PRODUCTION support:

* The test environments are hosted by Rackspace.
* The environment manifest will define which components get deployed into an environment.

# release Environment

## Release process to UAT



## Release notes

A release note for regular build includes the following content:

* Fixed bugs within this release
* Outstanding bugs
* New issues found if any

A release note for gold build includes the following content:

* The information about the release(build numbers, release numbers etc.)
* Details as of planned release – date and time
* New features added to this release
* All fixed bugs from previous release
* Fixed support tickets
* Outstanding bugs

# Entry and Exit Criteria

## Entry Criteria

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Type** | **Description** | **Owner** |
| **E001** | Entry | The scope of the release has been confirmed by the product owner. | Project Team |
| **E002** | Entry | The SYSTEST environment is available with reference data to allow testing. | Dev Team |
| **E003** | Entry | All development for the feature under test has been confirmed as ready to test (dev done). The code has been deployed to the SYSTEST environment. | Project Team |
| **E004** | Entry | The test analyst has been briefed by the developer and product owner and has appropriate supporting documentation. | Business Analyst |

## Exit Criteria

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Type** | **Description** | **Owner** |
| **E005** | Exit | No critical or high defects open at the end of system testing on features delivered to next test phase. | Project Team |
| **E006** | Exit | Test coverage is over 85% | Project Team |
| **E007** | Exit | Exploratory test session logs signed off by TM | Test Manager |

# Testing Tools

|  |  |  |
| --- | --- | --- |
| **Tool** | **Requirements** | **Usage** |
| **JIRA** | X&X instance of JIRA projects for each sprint team. | JIRA Bug/Issue management  Management Reporting |
| **XRAY plugin for JIRA** | Licensed version integrated with X&X instance of JIRA | Create/execute tests in JIRA  Inbuilt test execution reports |
| **Confluence** | XXXXXX confluence for X&X with separate client spaces | Used for collaboration & sharing artefacts |
| **Putty** | Downloadable & configurable resource | Accessing server logs & configuration files |
| **DbVisualiser** | License Key | Validating / inspecting data in databases using SQL |
| **Remote VMWare or Desktop** | Remote machine of desktop pre-configured with the above toolset | In-sprint Test Preparation and Execution |
| **Notepad ++** | Download free version | Used for reviewing interface and file formats. |
| **AppCheck & BurpSuite** | License Key | Perform security testing of websites |

.

## Defect Management

XXXXXX’s teams will use the X&X JIRA instance to track defects and for any scripting required for system testing.

The XRAY add-on for JIRA will be used by XXXXXX to manage their testing on the XXXXXX JIRA instance. XRAY is used for:

* Test creation and maintenance.
* Test scheduling and prioritisation.
* Test execution.

# Tracking and reporting

|  |  |
| --- | --- |
| **In-Sprint Testing** |  |
| **Test Reporting** | * Release notes will be provided for each UAT deployment or hand over. * JIRA dashboard will be used to report the test progress. |
| **Defect Reporting** | * As any critical or high severity defects are encountered, testers work with developers for getting them resolved. * This allows these defects to be fixed and retested as they arise, thus clearing the pipeline as we progress. * Any medium or low severity defects are recorded, prioritised and follow the JIRA workflow. * Between-Sprint Defect Reporting: Defects can be found in sprints on features that were ‘Done’ in previous sprints: These defects will be managed formally through JIRA workflows; These should go to the project backlog and get assigned to the appropriate Sprint backlog. * In-sprint Triage: Sprint teams have an unbounded access to a product owner to review defects found in testing. On-the-spot triage is therefore the most suitable method in-sprint. |
| **Management Reporting** | * Testing will form part of the management reporting at appropriate stages. * JIRA will be used to provide extracts for all defects reporting. * Other metrics can be added as the delivery pipeline progresses to out-of-sprint testing phases -   + Defects in severity order by business drivers/ functional area   + Cumulative Defect Analysis – Found vs.Status. This should give a view of the status of out of sprint defects found in a particular test phase (easily identify number of defects that were prioritised for fix by XXXXXX and actually fixed).   + Root cause analysis – Provide a measure on underlying cause of the defect with a view of identifying learnings and improvements for in-sprint testing. |