**Purpose of this document**

* Demonstrates the extent to which the tester has understood the nature and scope of the project
* Forms the contract between the entire project team regarding what is going to be tested and how it is going to be tested.
* Outlines how the project team knows it has met its goals

**Testing checklist**

These are the questions the tester must ask:

* Is every piece of development in the project testable?

How? If not, does it matter?

* What is the appetite for risk on this project?

How is it defined?

* Will the existing regression test pack be affected?

How?

Non-functional testing

* Data Volume testing
* Load testing
* Stress testing
* Browser compatibility testing
* Screen resolution testing
* Operating system testing

Contents

[1. Project Background 3](#_Toc391636365)

[2. Test Approach 3](#_Toc391636366)

[3. Scope of testing 4](#_Toc391636367)

[3.1 Acceptable level of Risk 4](#_Toc391636368)

[3.2 Functional Testing 4](#_Toc391636369)

[3.3 Regression Testing 4](#_Toc391636370)

[3.4 Test Automation: 4](#_Toc391636371)

[3.5 Non-functional testing (In progress) 5](#_Toc391636372)

[3.6 Cross Browser Compatibility Testing(In progress) 5](#_Toc391636373)

[3.7 Other Adhoc Testing Request 5](#_Toc391636374)

[4. Out of scope 5](#_Toc391636375)

[5. Risks and issues 5](#_Toc391636376)

[6. Test data 6](#_Toc391636377)

[7. Assumptions and Exclusions 6](#_Toc391636378)

[7.1 Assumptions: 6](#_Toc391636379)

[7.2 Exclusions: 6](#_Toc391636380)

[8. Test environments 6](#_Toc391636381)

[8.1 Testing environments 6](#_Toc391636382)

[8.2 Testing Tools 6](#_Toc391636383)

[9. Considerations when estimating 6](#_Toc391636384)

[10. Testing Management 7](#_Toc391636385)

[10.1 Project reporting 7](#_Toc391636386)

[10.2 Defect Reporting (Need to discuss and agree this flow) 7](#_Toc391636387)

[10.3 Defect Life cycle(Typical defect flow in Jira system)-this will be updated based on jira flow.(TBD) 8](#_Toc391636388)

[10.4 Roles and Responsibilities 8](#_Toc391636389)

[10.5 Test Deliverables 8](#_Toc391636390)

# Project Background

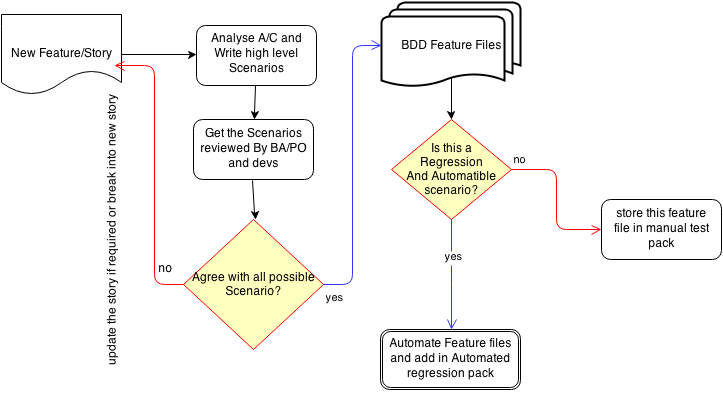
The goals of the HTML5 Getamap project are to replace the current getamap application at <http://www.getamap.ordnancesurveyleisure.co.uk/> with an HTML5 version that can be viewed by desktop and mobile browser-based clients

# Test Approach

The test approach in this project will purely follow BDD methodology. It’s highly recommended to use BDD tool to fulfil this testing approach.

Main objective is to have a high level of automation to support the proposed deployment process and also support regression testing. All possible functional tests will be automated with a BDD/TDD approach using Cucumber and Selenium web driver.

**Story life cycle with a BDD approach**.



In Jira under each story, Test analyst will create test tasks for performing required test functions as mentioned in above diagram. All efforts are to be recorded in Jira under stories.

During the sprint, Tester is supposed to involve in sprint planning to provide test estimations, deriving test requirements, testing scope, testability criteria and any environmental related issues to be discussed beforehand.

**Build deployment on OS environments:**

The suggested approach is to have automated build deployment through Jenkins server.

This means on build deployment on Jenkins, it will first execute unit tests (will be provided by Intelliscan) and on all pass, it will trigger smoke Test( this will be provided OS,Nasir Khan). Once all the tests are passed it will deploy the new code on CI environment.

If in case of build failure, Intelliscan may have to look into and fix the unit tests or functionality.

The Business Impact Assessment section of this document provides key input for the System Architecture Definition (covered separately) and for the Business Implementation Strategy in the second part of this document.

Where there is a need to change or re-enforce particular cultural aspects of the business describe current and future states clearly justifying the change.

# Scope of testing

## Acceptable level of Risk

This project will take a risk-based approach. This means that the objective is to ensure all ‘must-have’ requirements work as required with no service-affecting defects. This also means that some minor defects, known or unknown, may go into production. This is the acceptable level of risk

## Functional Testing

**Features to be Tested**

1. Data transformation & loading - migration of all live data which currently exists within GAM.
2. Verification of current GAM functionality implemented into the new solution
3. Verification of any new functionality
4. Account creation & registration of new customers
5. Account access for existing registered users
6. Customer preferences
7. Verifying user profiles & restrictions for new and existing customers. i.e. Guest users/Registered users/Subscribers
8. Expiry of user subscriptions
9. Standard UI functionality
10. Creating & editing routes (new & old)
11. Exporting & importing of routes
12. Purchases
13. Integration with Magento including registration, user authentication, payments
14. Printing
15. Browser compatibility for desktop and mobile devices
16. System interfaces - internal /external API's
17. Verification of performance related to NF requirements
18. Security

## Regression Testing

This is most critical test deliverable which is to be created and maintained as test harness. At the end of each sprint, as part of test closure activity need analyse test scenarios which can be re-run as regression to conform existing functionality is not broken. This would give a level confidence in the system with new iterative requirements.

## Test Automation:

The basic strategy for testing the new features or output of development will be a set of defined test cases/feature files for manual execution. In parallel to creating and executing manual test cases, we will need to create a regression test set, where possible these tests should be automated to execute using Selenium Web driver.

Test analyst is expected to provide estimates for test automation during sprint planning and will be included as “definition of done for the story.

Also mainly to support test automation, test analyst will require to setup a lightweight framework in Selenium Web driver. This may require some amount of time initially, Test analyst is expected to give estimates for this and may require to add a separate story for this.

## Non-functional testing (In progress)

Test analyst is expected to take sufficient inputs from the Architects and business analyst before planning for performance test.

Inputs such as

Expected response time,

Number of concurrent users

Stress levels

|  |  |
| --- | --- |
| **Testing task** | **Comment** |
|  |  |
|  |  |
|  |  |

## Cross Browser Compatibility Testing(In progress)

* + Cross browser compatibility testing will have an element of automation based on functionality only, using the SI/SIT automated test suite.
  + Any cross browser automation will be achieved using the Selenium framework, and will cover the OSX & MS Windows 7 OS's, and each with a set of multiple browsers & versions.

## Other Adhoc Testing Request

|  |  |
| --- | --- |
| **Testing task** | **Comment** |
|  |  |
|  |  |
|  |  |

# Out of scope

*Unit Testing will be done by developers only before committing their code into our repository.*

*Need to think of other areas. (In progress)*

# Risks and issues

|  |  |  |
| --- | --- | --- |
| **Risk/Issue** | **Comment** | **Action taken by team** |
| Environment Risk | Build issue or Jenkins | Hosting team may have to support |
| Testability Risk | Test data lack or Stubs lack | Business team: Provide test data  Dev Team: Support testers by providing stubs if required. |
|  |  |  |

Describe here the strategy for transitioning the organisation from its current state to the state encapsulated by the Business Vision section of the Business Foundations. Where possible use SMART descriptions (in this context: Specific, Measurable, Action-oriented, Realistic, Time-based). Consider:

Training staff in the operation of the new business processes

Outline the anticipated need for and extent of training required

Consider all classes of user

Consider all types of training from how to operate a new system to the wider cultural changes required

Where appropriate, consider training in the whole business process and not just the system part of it.

Education of End-users of processes and systems who are not part of the business organisation e.g.

Customers

Suppliers

Service Partners

# Test data

Product Owner may assist in getting test data from required sources to validate business level scenarios

Developers may have to assist with test data to support component level test data.

Technical coordinator may have to assist with different set of map data.

# Assumptions and Exclusions

## Assumptions:

Unit Testing is done by Developers before declaring any task/story ready for test

Details from developer for testing a Technical Task .

* QA will be given enough bandwidth (time) after making release on QA environment for testing new items in the build. Eg. No late builds on last day of sprint.
* QA environment would always up and running without any downtime, especially during major releases or end or sprint.
* Team/management might not expect all features to be automated by end of the sprint. However this will be QAs responsibility to automate in following sprint depending on time availability. This means Stories cannot be blocked if automation task is pending, still can be release to staging/UAT provided all test scenarios are manually verified and no issues.

## Exclusions:

Regression Testing may not be done couchbase system, however some areas which is part of end to end testing is done.

# Test environments

## Testing environments

The following environments are mandatory:

- Continuous Integration (for daily testing)

- QA/Testing Env

- Staging

*Developers need to setup stubs for component level testing in test environments if required.*

## Testing Tools

All test cases will be written in BDD feature file format and maintained in svn repo.

Automation Tool: Test automation scripts will be written in Java as JUNIT test cases using Selenium Web Driver and cucumber-jvm open source tools. Details how to setup will be provided in a separate document.

Goal for automation is to test new builds continuously on Jenkins which is expecting at later stage when we have end to system functionality.

All automation scripts to be maintained

# Considerations when estimating

Estimates provided for each requirement include:

* Test analysis and design
* Writing tests
* Creation of test data
* Executing tests against CI and QA builds
* Raising defects
* Re-testing
* Support for business users in their UAT testing
* Creation of automated tests where applicable
* Update of impacted existing automated tests
* Running all smoke and automated regression tests in each time box to ensure they all pass
* Any regression testing required

# Testing Management

## Project reporting

A test report will be produced at the end of each sprint and should be sent to the project management and posted to project confluence, visible to everyone in team involved in the project , detailing the regression tests and new tests executed at the end of each sprint.

Test reports should be provided at the end of any major phase or activity of the project, such as major release Additional test reports should be created and supplied as necessary to indicate project progress and according to project. Project test reports templates can be found on confluence.

## Defect Reporting (Need to discuss and agree this flow)

Issues identified in testing will be reported through JIRA in the area defined for the project, either as a new bug or by reopening the development story that has failed testing according to the following rules

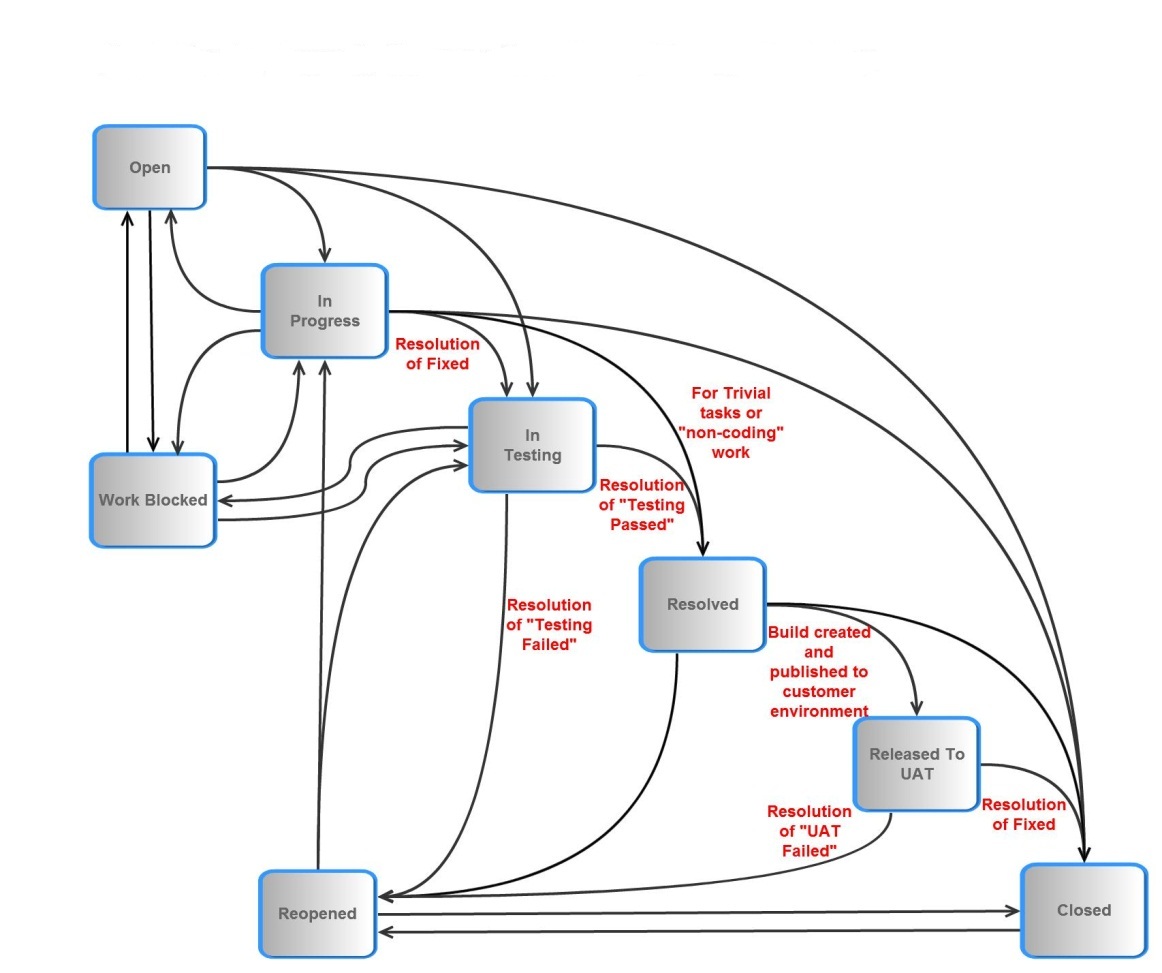
1. If a story under test fails some aspect of testing then the story should be set to ‘Blocked’ and a defect raised and assigned to the last developer to work on the story (or task), subject to discussion with the developer. This is the case where the failure is fundamental to the story and needs to be fixed for the story to be accepted as done. This should be the default behaviour.
2. Is the story catastrophically fails in testing then it may be appropriate to return the story itself back to the development team as it is clear that it is likely that the story has fundamentally been incorrectly implemented.

Alternative scenarios

1. If a defect is detected that relates to another story that was developed in the current sprint (but has already been marked as ‘Resolved’, i.e. testing passed), then the story should be reopened and assigned back to the last developer. This is the case where the failure is fundamental to the story and needs to be fixed for the story to be accepted as ‘done’.
2. Any faults detected relating to stories in the current sprint that we decide to not fix in the current sprint should be raised as new JIRA Bugs and assigned to a future sprint (e.g. a minor severity issue detected on the last day of the sprint or in the demo). We don’t want to carry over whole stories that have a minor defect associated with them.
3. If a defect is detected that relates to a piece of functionality (Story) completed in a previous sprint then a new JIRA Bug should be opened, as this is a regression in previously working software.

Eminently some common sense is required in judging if the failure observed is related to the story under test or not, with the intention to make sure that JIRA tickets are not kept open as new minor issues are detected that become unrelated to the original issue. Also we should really try to fix all faults relating to stories in the current sprint however small.

## Defect Life cycle(Typical defect flow in Jira system)-this will be updated based on jira flow.(TBD)



## Roles and Responsibilities

* **Development Testing (Unit Testing):** Will be the responsibility of developers.
* **System Testing**: Will be the responsibility of the test analyst and where needed by other development team members depending on the bandwidth. System testing will include all test analysis, test case creation, test code creation and maintenance and execution
* **Acceptance:** The project manager/Product owner will be responsible for accepting stories. It is assumed that acceptance will happen within one week following release of the story to the Business Analyst for User Acceptance Testing (UAT). It is assumed that BA or PO will engage in UAT prior to accepting a story, but setting the JIRA story to closed.
* **Test reporting**: Will be the responsibility of the QA Lead.
* **Test Strategy and Test Project management:** The Test Lead will be responsible for producing and maintaining a project test strategy.

## Test Deliverables

* User stories test Scenarios
* Test cases-feature files
* Test scripts(Automate new features)
* Test Reports (end of sprint, end of phase, to be stored on confluence)
* Automation test code(extending framework)