

RIFS-Test Strategy



Department for Business Innovation & Skills

Test Strategy

Grants Funding Programme

Author: Nasir Ali Khan
Date: 12/05/2016
Version: 1.0 [Initial draft]

Contents

- 1 Purpose of this Document:
- 2 Test Objective
- 3 Test Approach
- 4 Scope of testing
 - 4.1 Acceptable level of Risk
 - 4.2 Test levels
 - 4.2.1 Story Acceptance Test
 - 4.2.2 System/Regression Testing
 - 4.2.3 Exploratory Testing
 - 4.2.4 Test Automation:
 - 4.3 Non-functional testing (In progress)
 - 4.3.1 Load Testing:
 - 4.3.2 Accessibility Testing <Need to gather input on this>
 - 4.3.3 Cross Browser Compatibility Testing (In progress)
 - 4.3.4 Security Testing
- 5 Product Backlog Grooming
- 6 Sprint Planning
- 7 Build deployments
- 8 Out of scope
- 9 Risks and issues<need to understand more about the project to see what level of risks involved. Below are the general ones
- 10 Test data
- 11 Assumptions
 - 11.1 Assumptions:
- 12 Test environments
 - 12.1 Testing environments
 - 12.2 Testing Tools
- 13 Testing Management

- 13.1 Project reporting
- 13.2 Defect Reporting (Need to discuss and agree this flow)
- 13.3 Defect Life cycle(Typical defect flow in Jira system)-this will be updated based on jira flow.(TBD)
- 13.4 Roles and Responsibilities within delivery team
- 13.5 Test Deliverables

Purpose of this Document:

Forms the contract between the entire project team regarding test approach and how it is going to be tested.

Test Objective

Testing objective is to delivery constantly working and quality builds that meets business requirements by means of providing fast feedback and defect prevention and visibility at early stage of development.

Test Approach

As core agile principle, testing is not a separate phase but part of development. This will also give opportunity for team to support test by means of exploratory testing, usability testing.

Considering above agile test principle, test approach in this project will purely follow BDD and TDD methodologies. It's highly recommended to use BDD tool to fulfil this testing approach in order to perform automation testing.

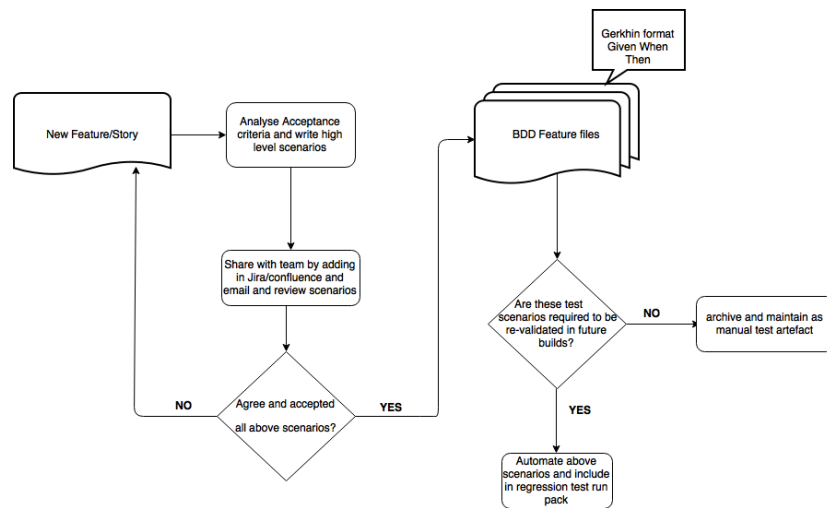
Implementing BDD approach in this project means understanding the business requirements at story level as acceptance criteria and translating as Test scenarios in plain English language in gherkin format.

These test scenarios would be outcome of short amigos within development and service designers to understand the requirements in details before actual development

Eventually agreed test scenarios to be automated which is implementing TDD process to ensure tests are written whilst story is being implemented at code level.

See below for detail story life cycle within a sprint.

Story life cycle within BDD approach.



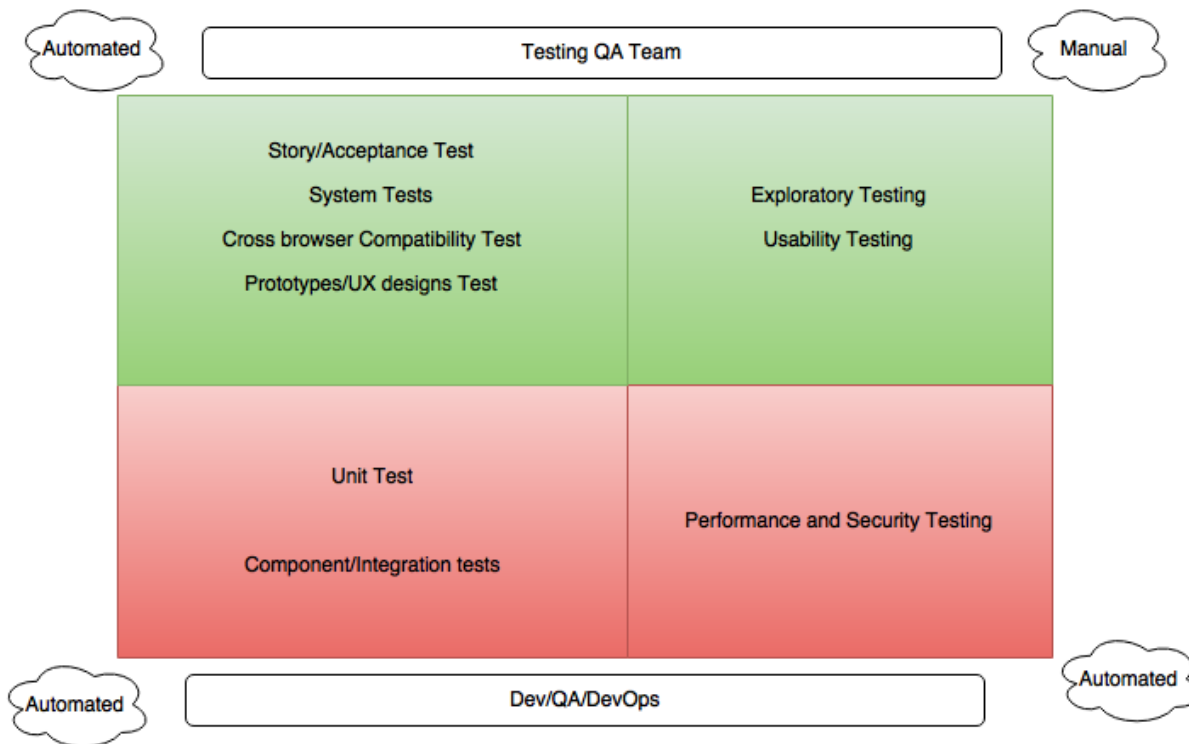
In Jira under each story, Test analyst will create test tasks for performing required test functions as mentioned in above diagram. All efforts are to be recorded in Jira under stories.

Scope of testing

Acceptable level of Risk

This project will take a risk-based approach. This means that the objective is to ensure all happy path 'must-have' requirements work as required with no service-affecting defects. This is the acceptable level of risk.

Test levels



Story Acceptance Test

As in agile development, it is expected to focus on acceptance testing ensuring defined stories meet business requirements. This will also ensure early visibility on issues if any and get them fixed during the sprint itself.

Primary focus is to ensure happy paths working as expected and some amount of edge cases covered. This will help business achieve what is expected in given time and same time recording any known issues/improvements spotted.

System/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 it is required to maintain above acceptance tests as regression pack in order to run these tests as part of the system tests.

Primary goal to achieve with this system test is to ensure new features wont impact existing functionality.

To reduce regression test time and perform continuous testing within the sprints, it is expected to automate regression pack.

Exploratory Testing

In agile team, apart from QA team doing exploratory testing on regular basis, it is encouraged to involve whole team in performing exploratory testing. This would be a short session guiding participants what to test and examples to use. This would help in getting some valuable feedback from other fresh pair or eyes. Also this would uncover any usability issue.

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)

Load Testing:

It is expected to perform some code level performance test for an early visibility of any performance issue.

However, release level load testing can take place before major release to ensure that code level, infrastructure level is performed taking the in input as mentioned below.

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

Proposed tooling for load testing: Jmeter

Also DevOps needs to support for desired fully integrated environment to perform load testing and monitoring.

Accessibility Testing

[Accessibility - Access Needs](#)

Cross Browser Compatibility Testing

Cross browser compatibility testing will have an element of automation based on functionality only, using the regression automated test suite.

Any cross browser automation will be achieved using the Selenium framework, and will cover as per the GDS guidelines

<https://www.gov.uk/service-manual/technology/designing-for-different-browsers-and-devices>

It is expected to assess cross browser automation tools such as Browser stack, Sauce Labs and choose best tool.

Security Testing

Currently in the CI pipeline basic OSWASP security scanning test is being performed which is being managed by DevOps.

However, it may be expected to perform security test by third party (external vendor) before beta release. (TBD)

Product Backlog Grooming

It is expected for test analyst to involve in product backlog grooming, this would help

- Test analyst understand future requirements
- Team/PO to understand any impact on current system functionality
- Improve acceptance criteria in story by raising questions
- Test analyst to give early visibility based on any dependency or any issue.

Also help user story to be in good state as below

- "I" ndependent (of all others)
- "N" egotiable (not a specific contract for features)
- "V" aluable (or vertical)
- "E" stimable (to a good approximation)
- "S" mall (so as to fit within an iteration)
- "T" estable (in principle, even if there isn't a test for it yet)

Sprint Planning

As a very common practice in agile team to hold half a day workshop within team and cross functional team members to go through the stories

before committing for the sprint. Also an opportunity to clarify if stories are in good readiness state to deliver.
Test analyst is expected to be part of this workshop and ensure test estimations are given to give visibility of test effort required. This would help team to understand the work load and manage resource accordingly.

- 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

Build deployments

As the development following an agile approach in sprints and story level, it is quite important for to have Continuous Integration which will have new features constantly being committed by dev team.
Considering above continuous changes in CI environment, on each commit it is expected to run automated unit test, sonarqube and smoke test on each commit ensuring that build is stable before it is deployed to CI.
Also it is required further to run automated Acceptance test to run on daily basis ensuring business features not broken with new changes.
Test Analyst to propose a controlled test environment to perform story testing and full system testing and other required testing which will be listed in Types of Testing.
Please see below link for detail understanding on the current deployment process and tooling.
<https://bisdigital.atlassian.net/wiki/display/GF/CI+Narrative>

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

<need to understand more about the project to see what level of risks involved>. Below are the general ones

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.

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.

Assumptions

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 of sprint.
- Team 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.

Test environments

Testing environments

The following environments are proposed:

- Continuous Integration (for Continuous testing)
 - QA/Testing Env
 - Demo/Staging env
- Developers need to setup stubs for component level testing in test environments if required.*

Testing Tools

Functional Test tool: Selenium Webdriver, Cucumber-jvm, Eclipse

All test cases will be written in BDD feature file format and maintained in GitHub 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 and perform regression test on demand basis. Also extend test tooling to support cross browser compatibility tests.

Non-Functional Test tool: Jmeter, Browser Stack/Sauce Labs.

Testing Management

Project reporting

A test report will be produced at the end of each sprint and should be document on confluence and link is sent to the stake holders. Also Jenkins test results report will be available to team.

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',

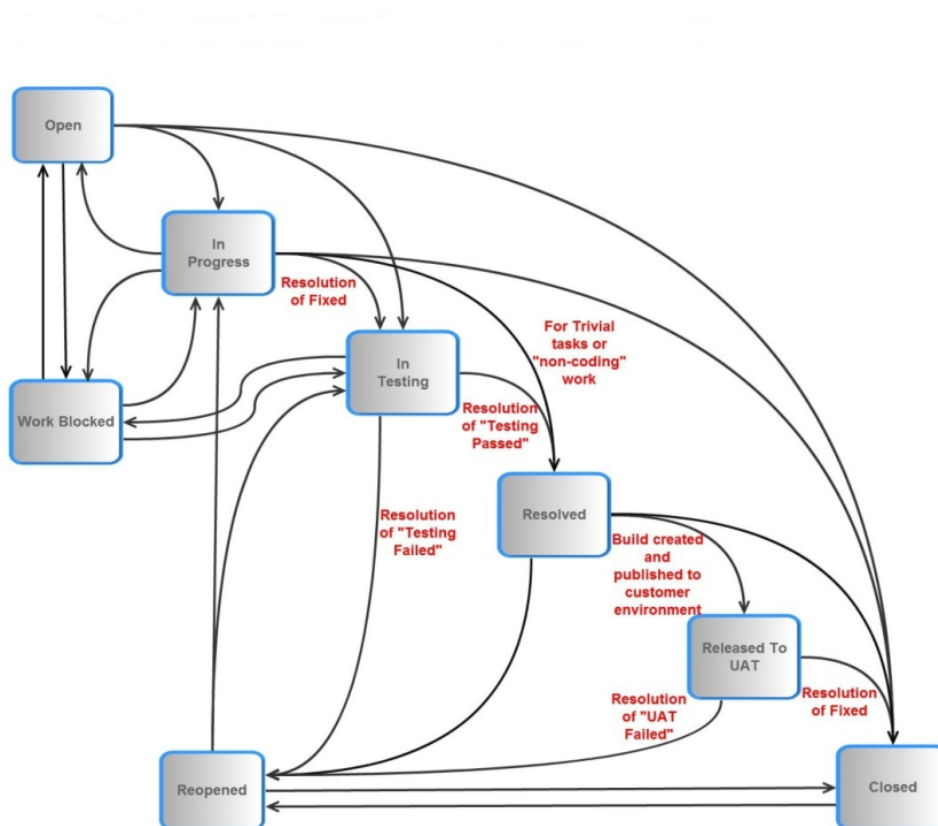
i.e. testing passed), then the story should be reopened and assigned back to the last developer. This is the case where the failure is fundamental to the story and needs to be fixed for the story to be accepted as 'done'.

- 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.

- 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 within delivery team

- Development Testing (Unit Testing):** Will be the responsibility of developers.
- Acceptance Testing:** QA team to perform Acceptance testing.
- 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
- Story/Feature sign off:** Product owner may want to have quick look before sign off or mark as completed and setting the Jira story as closed.
- Test reporting:** Test Lead to report testing outcome may be during the stand-ups, or sprint level and release level.

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)