

|  |  |
| --- | --- |
| Author | Chris Hustings |
| Date | 07/03/2019 |
| Version | 1.1 |
| Classification | Experian Internal |



**Solutions UK&I - Test Policy and Test Strategy**

# Document Control

Document Owner

|  |  |
| --- | --- |
| Name | Role |
| Jayne Barnaby | Director of Software Development (Responsible for Test Strategy and Test Improvement at SLT level) |

**Approval**

Following approval, this document will be base-lined by promoting it to a major version number (e.g. 1.0), recording the names and positions of approvers in the Approval Record and, if applicable, accepting all tracked changes. The Approval Record can be found in the Document Control Appendix at the end of this document.

A soft copy of email approvals will be maintained and stored as a permanent record of acceptance of this document.

**Change Control**

This document is subject to change control. All requests for change should be referred to the document owner. The Change History can be found in the Document Control Appendix at the end of this document.

Distribution

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Role | Reason | Organisation |
| Jayne Barnaby | Director of Software Development | Approver | Experian Solutions |
| Philip Moore | Director of Client Delivery and Service | Approver | Experian Solutions |
| Darren Bowles | Technical Consultant, Product Delivery | Review | Experian Solutions |
| Andrew Hoggard | Head of Development (CI) | Review | Experian Solutions |
| Meena Meenakshi | Senior QA Manager, Client Solutions | Review | Experian Solutions |
| Andy Bradfield | Test Governance Lead | Review | Experian Solutions |

**Contents**

Contents

[Document Control 2](#_Toc526176443)

[1. The Solutions UK&I Test Policy 5](#_Toc526176444)

[1.1 Quality First 5](#_Toc526176445)

[1.2 Represent the Customer 5](#_Toc526176446)

[1.3 Efficient Testing 6](#_Toc526176447)

[1.4 People and culture 6](#_Toc526176448)

[2. Solutions UK&I Test Strategy 7](#_Toc526176449)

[2.1 How the Test Strategy is structured 7](#_Toc526176450)

[3. How we Plan Testing 8](#_Toc526176451)

[3.1 Planning and estimation 8](#_Toc526176452)

[3.1.1 Updating Regression Packs and Automated Tests 8](#_Toc526176453)

[3.2 Planning during transition to the SAFe methodology 8](#_Toc526176454)

[4. When We Test 10](#_Toc526176455)

[4.1 Testing early in the development lifecycle 10](#_Toc526176456)

[4.2 Testing in the SDLC by phase 11](#_Toc526176457)

[4.2.1 Initiation 11](#_Toc526176458)

[4.2.2 Analysis and Design 11](#_Toc526176459)

[4.2.3 Build 11](#_Toc526176460)

[4.2.4 System Test 11](#_Toc526176461)

[4.2.5 Acceptance Testing 12](#_Toc526176462)

[4.3 Testing during transition to the SAFe methodology 13](#_Toc526176463)

[4.3.1 Sprint Planning 13](#_Toc526176464)

[4.3.2 During a Sprint 13](#_Toc526176465)

[4.3.3 Post Sprint 14](#_Toc526176466)

[4.3.4 Continuous Testing 14](#_Toc526176467)

[5. How we test 15](#_Toc526176468)

[5.1 Static Testing 15](#_Toc526176469)

[5.2 Risk based testing 16](#_Toc526176470)

[5.3 How we write tests 17](#_Toc526176471)

[5.3.1 Test Scenarios 17](#_Toc526176472)

[5.3.2 Test Cases 17](#_Toc526176473)

[5.3.3 Test Case Format 17](#_Toc526176474)

[5.3.4 Test coverage 18](#_Toc526176475)

[5.4 Create and Maintain Regression Tests 18](#_Toc526176476)

[5.5 Test environments and data 19](#_Toc526176477)

[5.5.1 Test Environments 19](#_Toc526176478)

[5.5.2 Test Data 19](#_Toc526176479)

[5.6 Test execution and test evidence 20](#_Toc526176480)

[5.6.1 Exploratory testing 20](#_Toc526176481)

[5.7 Defects and defect management 21](#_Toc526176482)

[5.7.1 Raising defects 21](#_Toc526176483)

[5.7.2 Defect triage process 21](#_Toc526176484)

[5.8 Suspension, resumption, entry and exit criteria 22](#_Toc526176485)

[5.8.1 Suspension and resumption criteria 22](#_Toc526176486)

[5.8.2 Entry and exit criteria 22](#_Toc526176487)

[5.9 Test Reporting 23](#_Toc526176488)

[5.9.1 Regular reporting 23](#_Toc526176489)

[5.9.2 Test Completion report 23](#_Toc526176490)

[6. Testing Metrics and KPIs 24](#_Toc526176491)

[6.1 Project Level Testing Metrics and KPIs 24](#_Toc526176492)

[6.1.1 Collated Metrics 24](#_Toc526176493)

[6.1.2 Derived KPIs 24](#_Toc526176494)

[6.2 Organisation / Service Level Metrics and KPIs 24](#_Toc526176495)

[6.2.1 Collated Metrics 24](#_Toc526176496)

[6.2.2 Derived KPIs 25](#_Toc526176497)

[7. Testing tools 26](#_Toc526176498)

[7.1 Test management tools 26](#_Toc526176499)

[7.2 Web service testing tools 26](#_Toc526176500)

[7.3 Functional automation tools 26](#_Toc526176501)

[7.4 Performance testing tools 26](#_Toc526176502)

[8. Team structure, roles and responsibilities 28](#_Toc526176503)

[8.1 Team structure 28](#_Toc526176504)

[8.2 Roles and responsibilities 28](#_Toc526176505)

[8.3 Culture of continuous improvement 29](#_Toc526176506)

[9. Exceptions 30](#_Toc526176507)

[9.1 Exception process 30](#_Toc526176508)

[Appendix A - ISTQB test definitions 31](#_Toc526176509)

[Appendix B - Document change history 32](#_Toc526176510)

# The Solutions UK&I Test Policy

The Test Policy describes the aims and behaviours of the Solutions UK&I testing groups and is divided into four main policy areas

## Quality First

* We will collaborate with all areas of the Solutions Business to agree common quality goals
* We will collaborate with our Clients to agree common quality goals and support their testing objectives
* We will engage with a project early in its lifecycle to ensure quality and testing is built into the planning and that the correct level and type of testing resource is included
* We will find defects as early as possible in the lifecycle
* We will Incorporate ISTQB definitions and common industry best practices

## Represent the Customer

* We will represent the whole end user experience with both functional and non- functional requirements being tested
* We will aim to minimise the number of issues an end user may see in production
* We will clearly communicate the measured quality of the software under test to our internal customers to support decision making
* We will represent the Independence of testing and fulfil that role in the triangle of influence within a project’s leadership group

## Efficient Testing

* We will ensure we focus testing in the most important areas
* We will avoid duplication of testing effort between teams
* We will use appropriate tools to support testing with automated solutions being introduced wherever possible

## People and culture

* We will use dedicated testing professionals in conjunction with the wider delivery teams to achieve the best possible test outcome
* We will work with our strategic partners to make best use of our groups skills
* We will work in a culture of continuous improvement with regular reviews of our strategy and processes

# Solutions UK&I Test Strategy

## How the Test Strategy is structured

The Test Strategy describes how we meet the aims of the Test Policy described in the previous section.

How the policy maps to sections in the strategy is shown below

# How we Plan Testing

## Planning and estimation

Planning and estimation will take into account both functional and non-functional testing requirements, and supporting activities such as creation and configuration of environments and data selection and creation.

The estimation model will include all the standard test items as defined by the Solutions SDLC RACI which are in summary;

* Static Testing Report
* Master Test Plan
* Test Cases
* Test environment setup and configuration
* Test data identification/creation
* Executable Test Scripts (Scripts for API testing, automation etc.)
* Test Execution (Including defect retesting)
* Test Completion Report
* Updating Regression Packs and Automated Tests

### Updating Regression Packs and Automated Tests

The need to maintain existing regression tests and automated tests should be assessed against changes or additions being made by the project.

This is to ensure that subsequent projects will have up to date tests to utilise and where automated testing and build capabilities exist, they will continue to function post project delivery.

In addition, it may be required for a project to create new regression packs and to automate existing regression packs for the Product being changed by the project.

This will be considered in the early planning process for a project and will be conducted in conjunction with the Product Owners and Project Team.

## Planning during transition to the SAFe methodology

As Solutions transitions to a Scaled Agile methodology the actual deliverables will be adapted to work with Stories and Sprint or Kanban boards at the team level.

A Master Test Plan document will be maintained at the Programme/Project level but will no longer contain elements such as a detailed test schedules, as this will now be covered in the sprint planning events during the sprint cycles.

Many of the same tasks will still be relevant during Sprints but with a greater emphasis on adoption of automated testing, both by developers and testing specialists in the Scrum teams.

# When We Test

## Testing early in the development lifecycle

Testing is not just running tests against the final delivered system, testing occurs in multiple forms throughout the software development lifecycle. The aim of this strategy is to identify issues as early as possible which reduces the overall cost to resolve issues and mitigates the risk of late discovery impacting releases. The various types of testing applied in the standard Solutions project delivery process is shown below

## Testing in the SDLC by phase

This section describes in more detail the testing that takes place where the Testing Teams have direct involvement, areas such as Unit testing which is largely a development team exercise is not detailed here.

### Initiation

* During the Initiation phase we will apply the Static Testing to the requirements. This is an exercise performed jointly with the business analyst/requirement owners and the testing team.
* The Master Test Plan is also written during this phase with the test scope, environment and data requirements, risks and any variations from the Test Strategy being agreed.

### Analysis and Design

* During the Analysis and Design phase we apply Static Testing to the next level of requirements such as functional specifications, this is an exercise performed jointly with business analyst/requirement owners and the testing team.
* At the end of the Analysis phase we will risk assess requirements.

The risk assessment is performed jointly with the Product Owner and senior development Representatives.

### Build

* During the build phase we will review design documents to aid with our understanding of the solution and we will also liaise with the development lead to understand the outcome of unit and component testing to further direct which areas of the system we will focus our system testing.
* During the build phase we write test cases to meet functional and non-functional requirements and prepare test data
* The Master Test Plan is also completed during this phase adding detail around test schedule.

### System Test

* During the System Test phase, we execute our test cases, recording results, collating evidence and raising defects.
* Defect retesting also takes place during this phase.
* Regular progress reports are produced during this phase.
* At the end of this phase the Test Completion Report is written.

### Acceptance Testing

* The testing teams are sometimes employed in a support capacity during business or client acceptance testing, helping with issue triage and with defect recreation and retesting. The level of involvement varies depending on the size of the Client and the type of project in acceptance.

## Testing during transition to the SAFe methodology

As Solutions transitions to a Scaled Agile methodology, the whole Analysis and Design, Build and System Test phases will be running as a series of Sprints as part of a larger release train.

Testing will take a number of forms before, during and after the Sprint Cycle itself

### Sprint Planning

* Static Testing will be applied to new Stories in the backlog. This will happen prior to Sprint Planning to enable accurate estimation
* Stories will also be risk assessed before or during Sprint Planning.

### During a Sprint

* Developers will utilise Test Driven Development to cover Unit and Component Testing activities.
* Automated tests will be utilised by the development team to confirm their builds are functioning.
* Functional tests will take place based on story acceptance criteria, these will be automated wherever possible and may be supported by some manual test cases and exploratory testing for areas that are difficult to automate. This will typically be performed by a testing specialist within the sprint team.

### Post Sprint

* At the end of a series of sprints when a whole solution is available, a period of end to end Functional Testing and Acceptance Testing may be utilised with a mixture of sprint team members and Client or Business Representatives.

### Continuous Testing

* In addition to sprint focussed testing it is envisaged that a continuous test cycle will be utilised, this will have aspects such as automated functional regression packs to ensure ongoing quality, performance testing to make sure service is not impacted and security scans to ensure security compliance. This will be integrated into the continuous build and release cycle which is planned to be implemented by the DevOPs team.

# How we test

## Static Testing

We will routinely perform Static Testing of requirements. This process is designed to identify defects related to requirements early in the software lifecycle and thereby reduce costs of defect resolution in later phases.

This takes the form of a standard set of questions to be applied against each requirement

The questions cover the following key attributes of a requirement

A Static Testing Report will be produced and defects will be raised in JIRA to track the resolution of issues identified during this phase.

As we transition to the SAFe methodology Static Testing will take place at the Story level and will be integrated into our productivity tool e.g. JIRA.

## Risk based testing

We will utilise risk based testing to make sure we run the most important tests first and to focus testing efforts in the areas with the most risk.

The risk assessment is conducted on detailed business requirements which are represented in JIRA by the Story Issue Type. These are typically extracted from the functional specification but may also be drawn from other sources such as direct creation of Stories for Agile projects.

The assessment is performed in conjunction with a business representative, who provides the impact of failure and technical representative who provides the likelihood of failure these are updated directly in JIRA with the risk rating being a calculated field.

The risk assessment process uses the following standard grid.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **Impact of Failure** | | | |
|  |  | Minor | Moderate | High | Significant |
| **Likelihood of Failure** | Very Likely | P3 | P2 | P1 | P1 |
| Probable | P4 | P3 | P2 | P1 |
| Possible | P4 | P3 | P3 | P2 |
| Remote | P5 | P4 | P4 | P3 |

By default, all requirements are deemed to be in scope for testing, the risk assessment can facilitate the process of limiting the test scope to match the schedule, cost and risk appetite for a given project.

The test scope will be agreed with the project leadership team (Project Manager, Development Lead, Test Lead and Senior Business Analyst) and the Business Stakeholders for the project.

## How we write tests

### Test Scenarios

Test Scenarios are intermediate level test design artefacts which are used to broadly check test coverage and to help with project estimation.

They are typically produced early in the project when forming the Test Approach and help guide the creation of test cases in later phases.

They are created in JIRA using the Test Scenario Issue Type.

### Test Cases

Test Cases are the lowest level test design artefact; these are represented in JIRA by the Test Issue Type.

Test cases should have enough detail to allow for regression testing to be performed and for automated tests to be written from them.

### Test Case Format

Test Cases will be written in JIRA with detailed steps to allow easy execution and automation

The basic layout of a Test Case is as follows.

|  |
| --- |
| Test Case Level Fields |
| Summary – A short description of the purpose of the Test, this appears in lists within JIRA. |
| Description – A detailed description of the purpose of the tests and any supporting information |

|  |  |  |
| --- | --- | --- |
| Test Step | Test Data | Expected Result |
| First step usually contains any required setup/ prerequisites | Details of pre-requisite data or settings |  |
| Subsequent steps will contain the detailed steps to perform | Details of intermediate Input Data | Details of the intermediate steps Expected states. |
| Final step is usually the main purpose of the test | Details of Input Data | Details of the Expected Outputs |

### Test coverage

Test cases will be written to meet both functional and non-functional requirements, and will cover both positive and negative conditions.

The number of test cases for a given requirement (the depth of testing) is linked to the risk priority of the requirement.

|  |  |
| --- | --- |
| **Requirement Risk Priority** | **Depth of Test Coverage** |
| P1 | Multiple positive and negative test cases |
| P2 | At least one positive and multiple negative test cases |
| P3 | At least one positive and one negative test case |
| P4 | At least one positive test case |
| P5 | Ad hoc test only / consideration for out of test scope |

Test cases will be stored in JIRA and linked to Stories so that test coverage can be demonstrated.

Test cases in JIRA will inherit the risk rating from their linked Story to enable Risk Based Test Cycles to be created.

## Create and Maintain Regression Tests

Regression tests will be created and maintained for all products that are actively being sold and which are also likely to have ongoing development requirements.

The level of regression coverage will vary from product to product and will depend on the relative business priority and the level of anticipated change, but in general will include;

A series of tests to be used as a smoke test for quick and efficient regression testing, these tests will form the basis of an automated smoke test pack for use in automated build processes and for code and environment acceptance tests.

A series of more in depth tests for key product functions will be created, these will be used for product changes and enhancements to make sure key functions are not affected.

These tests will likely be a mixture of manual and automated tests, and coverage will be built up as projects change and enhance product offerings.

High priority functional regression tests will be automated.

Regression tests will be held and maintained centrally in JIRA and will be linked to supporting code repository tools for automated testing code.

Tools may be used to help identify applicable regression tests for a given project.

## Test environments and data

### Test Environments

Test execution will take place in a controlled test environment separate to the development environment.

Access to the test environments will be limited as per Experian’s data and security policies with appropriate test data being selected and removed when no longer required.

Test Accounts will be requested via Service Now and will be granted for the duration of the intended project.

### Test Data

Test data requirements will be identified early in the planning process to ensure any test data that is required for a project is available or if it requires new data to be created.

Test data may be a copy of live data or obfuscated data depending on the business area and the project needs.

Use of live data for testing purposes will be subject to compliance and information security sign off on a per project basis where it is required. Sign offs should be sought during the initiation phases of a project. However, a particular restriction is that CIFAS data cannot be accessed by team members outside of the UK/EU.

## Test execution and test evidence

All test results will be recorded in JIRA which records the result, who executed the test case and when the result was updated.

Any failed tests will have a corresponding defect raised in JIRA which will be linked to the test result.

In addition to capturing the test result and any associated defects, test evidence will routinely be collected for all System Testing.

The evidence can take multiple forms depending on the type of testing, but is generally a screen shot of the state of the system under test, but may include log file extracts or database snapshots.

As we employ risk based testing a project can opt to only capture test evidence for the higher priority tests this should be agreed during the initiation phase of the project and recorded in the project test approach variations section when it is written.

### Exploratory testing

In addition to executing the prepared test cases exploratory testing will also be employed as a method to detect defects.

## Defects and defect management

### Raising defects

When an error is found, either in the system under test or in documentation a defect will be raised.

Defects will be raised in JIRA and will contain enough information to enable that defect to be reproduced. Defects will be classified by Severity (The Impact of the defect) and Priority (How quickly it should be resolved).

The table below shows the Severity and Priority values available in JIRA Global.

|  |  |
| --- | --- |
| **Severity** | **Priority** |
| 1-Critical | Blocker |
| 2-Major | Critical |
| 3-Average | Major |
| 4-Minor | Minor |
| 5-Cosmetic | Trivial |
| 6-Enhancement |  |

### Defect triage process

A triage process will be used when new defects are raised to ensure three things, to agree that the defect is valid, that it contains enough information and to agree the severity and priority of the defect.

The Test Lead will own the triage process which usually takes the form of a regular conference call or WebEx session.

The frequency of the triage is flexible depending on the size and defect activity of the project but daily is the ideal frequency to ensure defects are addressed in a timely manner.

For Agile projects the daily stand-ups and closer team working models will replace the need for formal triage sessions.

## Suspension, resumption, entry and exit criteria

In general, we will endeavour to test in a pragmatic fashion, working around issues we come across. However, there are times where this is not feasible and testing may need to be suspended.

### Suspension and resumption criteria

The following reasons may prompt a suspension of testing;

* A significant number of defects are found in early testing, particularly those of Critical or Major severity. This may be an indication of major development issues and further testing may not be of value.
* Environmental issues preventing successful test execution
* If requirement/design changes are identified during testing which would invalidate the current functional design.

Testing can resume when the issues have been resolved to the satisfaction of the Project Leadership Group (e.g. Project Manager, Development Lead, Test Lead and Senior Business Analyst).

### Entry and exit criteria

The standard Entry Criteria for the System Test phase are;

* Satisfactory evidence of Unit and Component Integration testing from the build phase
* System test environment is setup and configured
* No unresolved Critical defects exist from earlier phases
* No unresolved Major defects exist from earlier phases with the exception of those that have a planned resolution and timescale which will not immediately impact the testing progress.

Exit Criteria for the System Test phase are;

* The agreed testing scope has been completed
* No unresolved critical defects
* No unresolved major defects unless they have been Accepted, or have an agreed plan for resolution and timescale for resolution that does not impact the Acceptance phase.
* Approved Quality Gate Report

## Test Reporting

### Regular reporting

A regular progress report will be produced which shows test progress against plan and defect statistics, the frequency of the report should be agreed with the Project Manager.

The progress report will be circulated to the Project Leadership Group and selected business stakeholders.

### Test Completion report

This report will be produced at the end of the System Test Phase and will summarise all the testing up to that point, providing both commentary and statistics of the test results.

The report will also detail defect statistics including commentary on outstanding defects and their planned resolutions.

The approval of this report by the Project Leadership Group and the Project Stakeholders is a mandatory requirement to entering the Acceptance phase of a project.

# Testing Metrics and KPIs

Metrics will be routinely collected for all Solutions projects and will be used to derive Key Performance Indicators (KPIs) at both the Project and Organisation level.

The KPIs will be used to track things such as Test Effectiveness, Test Efficiency and Automation Uptake.

## Project Level Testing Metrics and KPIs

### Collated Metrics

* Number of Defects Found by Project Phase
* Number of Defects Found by Severity
* Number of Re-Opened Defects
* Number of Defects Found by Root Cause
* Time Taken for Test Case Creation by Type (Manual/Automated)
* Total Number of Test Cases by Type (Manual/Automated)
* Time Taken for Test Case Execution by Type (Manual/Automated)
* Total Number of Executed Test Cases by Type (Manual/Automated)

These metrics will then be used to derive the following KPIs.

### Derived KPIs

* Defect Leakage into UAT
* Defect Leakage into Production (During Early Life Support/Warranty Periods)
* Overall Defect Leakage (Test Effectiveness)
* Defects Found per Test Case Executed
* Test Case Design Efficiency (Cases per Hour)
* Test Execution Efficiency (Cases per Hour)
* Defect Re-Open Rate

## Organisation / Service Level Metrics and KPIs

In addition, the following metrics will be collected at the Organisation/Testing Service Level.

### Collated Metrics

* Total Number of Available Manual Test Cases
* Total Number of Available Automated Test Cases
* Total Number of Test Cases Executed by Type (Manual/Automated)
* Total Number of Defects Found

### Derived KPIs

* Percentage of Test Cases that are automated
* Number of Execution Cycles that are automated vs. Execution Cycles that are manual
* Defects found per Test Case Executed
* Test Case Design Efficiency
* Test Execution Efficiency

As automation take up increases, test execution efficiency should increase, measuring case design efficiency will allow us to monitor any impact of automated test creation and maintenance.

Additional metrics may be collected by local teams to support specific business needs but will not be mandated Solutions wide.

# Testing tools

We use tools to make our testing processes more effective and efficient. This section describes at a high level the tools we utilise and what we use them for.

## Test management tools

For test and defect management we use JIRA. The Zephyr add-on is specifically used for the Testing function. We also use JIRA to store requirements (in the form of stories), so that traceability can be demonstrated.

We use the Experian JIRA Global instance which is shared between a number of business units globally.

Some elements can be customised on a project by project basis but shared items such as workflows and common dropdown values such as severity and priority are fixed to allow common reporting and to allow Scaled Agile processes to work across Portfolios.

## Web service testing tools

We typically use SoapUI for the testing of services and APIs. This is planned to be replaced by SOA Gear, a API testing tool which will integrate with the Automated Testing Framework.

## Functional automation tools

We currently use Selenium (IDE and Webdriver versions) for automated functional testing of some of our web based products with various control interfaces being used to drive the scripts.

An automation framework is being developed to standardise this approach and to widen the coverage of the functional automation capability. This Framework will also be based on Selenium.

Some localised tooling may currently exist within the Solutions business with existing tests in place these may continue to run in those tools if migration to the new Framework is not cost effective.

## Performance testing tools

A number of techniques are used for performance testing some of which involve manual measurement of performance using captured system statistics such as elapsed times, MSUs used etc. JMeter has been used for measuring performance in certain projects.

A performance testing framework also based on JMeter is being developed so we have consistent methods and reporting available for Performance testing. The Framework will cover a variety of test types such as Load. Stress, Spike and Soak Testing.

# Team structure, roles and responsibilities

## Team structure

The Solutions organisation has a partnership with UST Global who provide Testing resources for projects. They work in both an offshore and onshore capacity

Solutions also have some Experian Testing staff in certain business areas and in Client Facing and Supporting Roles

## Roles and responsibilities

**Specific Experian Roles**

#### Test Office - Overall responsibility for the Test Policy and Test Strategy and for Test Improvement Processes, oversight of the Testing Service.

QA Manager (Client Delivery function) – Client Facing role for Major Clients, conduit and review point between UST and Client Teams, Management of QA Leads.

QA Lead (Client Delivery function) – Client facing role, conduit and review point between UST and Client Teams.

**Common Roles, Experian and Partner resources**

Test Manager – Management of Test Leads. Test Resourcing(UST), Level of Testing Service

Test Lead – Management of Test Analysts, Test and Defect Management of projects

Test Analyst – Test Design and Test Execution

Detailed roles and responsibilities descriptions can be found in supporting process documents for the various business areas.

## Culture of continuous improvement

As an organisation we are committed to deliver a culture of continuous improvement, this takes a number of forms

* We operate regular forums where people are encouraged to share ideas and knowledge
* We will capture Lessons learned as part of the Test Summary report which will be stored centrally in the test office repository so they can be applied to future Test Approaches
* We will add test scope to cover defects found during exploratory testing and will create and maintain reusable regression packs
* We will have formal periodic reviews of this Test Strategy to make sure it continues to meet our business needs.

# Exceptions

## Exception process

It is anticipated that the vast majority of projects will adhere to this Test Strategy in full but occasionally a project may need to seek an exception. Any exceptions from the Test Strategy at a programme or project level shall firstly be agreed with the Project Leadership Team and the business stakeholders for the project and will be recorded in the relevant project or programme Test Approach document.

Secondly the exception should be verified by the Test Office to ensure the risk of doing so has been clearly communicated and agreed by the project’s stakeholders.

Any risks due to exceptions will also be raised in the Project Workbook.

## Appendix A - ISTQB test definitions

Test Policy – A high-level document describing the principles, approach and major objectives of the organization regarding testing.

Test Strategy - A high-level description of the test levels to be performed and the testing within those levels for an organization or programme (one or more projects).

Test Approach - The implementation of the test strategy for a specific project. It typically includes the decisions made that follow based on the (test) project's goal and the risk assessment carried out, starting points regarding the test process, the test design techniques to be applied, exit criteria and test types to be performed.

Test Plan - A document describing the scope, approach, resources and schedule of intended test activities. It identifies amongst others test items, the features to be tested, the testing tasks, who will do each task, degree of tester independence, the test environment, the test design techniques and entry and exit criteria to be used, and the rationale for their choice, and any risks requiring contingency planning. It is a record of the test planning process.

Test Condition - An item or event of a component or system that could be verified by one or more test cases, e.g., a function, transaction, feature, quality attribute, or structural element

Test Case - A set of input values, execution preconditions, expected results and execution post conditions, developed for a particular objective or test condition, such as to exercise a particular program path or to verify compliance with a specific requirement.

Appendix B - Document change history

Change History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Author | Details |
| 4/7/2018 | 0.1 | Chris Hustings | Initial draft for new Solutions structure post UST transition and to account for transitional state to SAFe from traditional waterfall like methodology. |
| 6/7/2018 | 0.2 | Chris Hustings | Internal review with Andy Bradfield in Test Office |
| 10/7/2018 | 0.3 | Chris Hustings | Updates to Agile section after walkthrough with Jayne Barnaby. |
| 20/7/2018 | 0.4 | Chris Hustings | Updated policy, added additional text into planning section, around regression and automation, added graphics for certain sections (phases, static testing, risk based testing) |
| 30/08/2018 | 0.5 | Chris Hustings | Updates following reviews with Jayne Barnaby and Andy Hoggard.  Updates following session with Meena to reflect some Client specific changes and to align some wording for a Client facing version. |
| 25/9/2018 | 0.6 | Chris Hustings | Some minor updates following review with Darren Bowles and Rick Harrison (DevOPs) , and a few wording updates to make a few areas a bit clearer.  Removed reference to Test Approach, replaced with Master Test Plan as the single document to support the Strategy at project level (and to align with Client Solutions needs) |
| 1/10/2018 | 1.0 | Chris Hustings | Baseline version for approval |
| 07/03/2019 | 1.1 | Chris Hustings | Updated Data Classification to Experian Internal. |

Approval record

|  |  |  |
| --- | --- | --- |
| Major Version | Approved By | Organisation |
| 1.0 | Jayne Barnaby | Solutions Development |
| 1.0 | Philip Moore | Client Solutions |