



Post Office Limited Programme Test Strategy



Table of Contents

1	Document Specific Information	3
1.1	Overview	3
1.2	Document Identifier	3
1.3	Issuing Organisation	3
1.4	Approval Authority	3
1.5	Change History	3
1.6	Distribution	4
1.7	References	4
2	Executive Summary	5
3	Introduction	6
3.1	Background	6
3.2	Document Purpose	6
3.3	Objective of Testing	6
3.4	Scope	7
3.5	Out of Scope	7
4	Cross-Project Organisational Test Strategy	9
5	Programme / Project Test Strategy	10
5.1	Programme / Project Risk Management	10
5.1.1	Testing as a Risk Management Activity	10
5.1.2	Risk Based Testing	10
5.1.3	Management of Programme / Project Testing Risks	11
5.2	Programme / Project Test Approach / Methodology	11
5.3	Testing Lifecycle	12
5.4	Testing Sub-Processes (Test Types / Phases)	12
5.5	Test Selection and Prioritisation	13
5.6	Testing degree of Independence	13
5.7	Governance	14
5.8	Defect management	14
5.8.1	Defect Management Severity Definitions	15
5.8.2	Priority Code Definitions	15
5.8.3	Defect Management of Testing Completion	16
5.8.4	Defect Tooling	16
5.9	Test Documentation and Deliverables	16



Programme Test Strategy v<x>

5.10	Reporting and Metrics	16
5.11	Configuration Management of Test Work Products	17
6	Testing Resources	18
6.1	Test Tools	18
6.1.1	Test Management Tools	18
6.1.2	Defect Management Tools	18
6.1.3	Test Execution Tools	18
6.1.4	Test Automation Tools	18
6.1.5	Performance Testing Tool	18
6.1.6	Other Test Tools	19
6.2	Test Automation	19
6.3	Test Environments	19
6.4	Test Data	19
6.5	Testing Organisation	19
6.5.1	Key Roles, Accountabilities and Responsibilities	19
6.5.2	Test Team Structure	20
6.5.3	Staffing and Training Needs	20
7	Test Sub-Process (Test Type / Phase) Specific Information	21
7.1	Unit Testing	21
7.2	System Testing	22
7.3	System Integration Testing (SIT)	23
7.4	User Acceptance Testing (UAT)	24
7.5	Business Acceptance Testing (BAT)	25
7.6	Non-Functional Testing (NFT)	25
7.6.1	Performance Testing	27
7.6.2	Operational Acceptance Testing (OAT)	28
7.7	Security Testing	31
7.8	Regression Testing	31
7.9	Post-Implementation Testing	31
8	Risks, Assumptions, Issues and Dependencies (RAID)	32
9	Glossary	33



1 Document Specific Information

1.1 Overview

A brief overview of where this document sits in the hierarchy of the document suite, identify the document and explain origins and history.

Example: This document will provide the general test strategy to be followed across Post Office Limited and Third Parties.

1.2 Document Identifier

Reference ID or number assigned to the document, according to whatever process is used.

Example: POL Test Strategy v1.0

1.3 Issuing Organisation

Which team or function wrote / owns the document

Example: POL or KPMG Test Team

1.4 Approval Authority

Insert names and roles of approvers (appropriate stakeholder approvals should be agreed in advance)

Name	Role	Version	Approval Date
<i>Simon Oldnall</i>	<i>Approver</i>		
<i>Harshwardhan Soman</i>	<i>Approver</i>		
<i>Benjamin Romberg</i>	<i>Reviewer</i>		

1.5 Change History

Must be updated as new versions are issued

Version	Author	Date	Description
0.1	<i>Chris Laughlin</i>	<i>24/02/2021</i>	<i>Initial draft</i>
0.2			<i>Internal review</i>
0.3			<i>Project review</i>
			<i>Project review</i>
			<i>Last updates added</i>



1.6 Distribution

Insert names and roles of people (or teams) that the document has been issued to

Name	Role	Version	Issue Date
<i>Simon Oldnall</i>	<i>POL – IT Director</i>		
<i>Harshwardhan Soman</i>	<i>POL – Testing and Release Mgr</i>		
<i>Dan Addy</i>	<i>POL – Head of HZ Architecture</i>		
<i>Martin Godbold</i>	<i>POL – Head of HZ Live Service</i>		
<i>Dean Bessel</i>	<i>POL – HZ Risk, Security&Investigats</i>		

1.7 References

Information on documents referred to in the document (in particular, the Organisational Test Strategy!)

Reference	Document	Version	Location
[1]	<i>POL Test Policy</i>	<i>1.0</i>	<i>Testing project folder</i>
[2]	<i>FJ COMMGTREP4166 - TESTING-QA</i>	<i>1.0</i>	
[3]	<i>FJ COMMGTREP4168 - SDLC</i>	<i>1.0</i>	
[4]	<i>FJ COMMGTREP4169 BED Report</i>	<i>1.0</i>	
[5]	<i>Environments list</i>	<i>1.0</i>	
[6]	<i>Project RAID</i>	<i>1.0</i>	
[7]	<i>POL Change management</i>	<i>1.0</i>	



2 Executive Summary

One-page summary of the document, including key points and signalling content and format of document.

Example: This document intends to define the approach to be followed during the Software Testing Lifecycle (STLC) to achieve the project test objective(s) and mitigate the risks before the final product reach its completion.



3 Introduction

3.1 Background

The background should give a short summary of the programme or project that this test strategy will cover, and how the scope of this document is supported by the Organisational Test Strategy.

Include information on the technical elements of the solution (if appropriate), identifying key risk areas

Example: KPMG LLP ("KPMG") are currently engaged by the Post Office Limited ("POL") to help with a range of issues pertaining to Horizon remediation efforts. Work commenced in October 2020 and was planned to end in December 2020.

Two streams of work were contracted. The first was assistance with defining investigations and forensics target operating model for Horizon IT. Additional requests were added to this scope and these are planning to wrap up mid-December 2020. The second was to conduct an audit of key areas of Horizon. This work was predicated on having access to the Horizon 3rd party service provider ("Fujitsu" or "FJ").

The focus of work was instead shifted to go deeper on issues through a POL lens. This work is ongoing with an interim report on it second iteration, being issued in December 2020.

Additional assistance was also requested with forensic examination of an IT terminal and we understand there to be need for more assistance in this domain. This is being managed through a formal Change Request.

3.2 Document Purpose

The purpose of this document is to describe the overall scope and approach to testing that will be applied to the <name of programme or project>.

Example: for this project all the software development will be done by Fujitsu third party as well as Unit Testing and System Testing. Any other testing activities will be done by Post Office.

3.3 Objective of Testing

What are the key risks that need to be tested on this project, and why.

Example: Testing will be looking to de-risk the key business processes involved on the changes performed on Functionality, Products, Processes, Accounting, Technology and Infrastructure. In the past POL and third parties weren't following a clear STLC.

The following section describes the testing objectives set out for this project:

- *Validate through testing that the solution designed and built supports the business processes*
- *Validate through testing that the developments and configuration introduced meet business processes*
- *Validate through testing that Horizon integrates with legacy systems and the solution functions as expected in an integrated landscape*
- *Validate through testing that the reports introduced by this project function as expected*
- *Validate through testing the controls, roles and authorisations introduced by this project are compliant*
- *Validate through testing that the system performs with agreed reliability standards*



Programme Test Strategy v<x>

- Use migrated data to run and validate end to end business processes
- Gain business acceptance of the implemented solution through UAT

3.4 Scope

Document what is in scope for this strategy – what changes are being delivered by the programme / project, what needs to be tested, and what types of testing are required?

Document what is in scope for this strategy – change types, projects / portfolios, areas of the organisation / business, test types etc.

Examples:

Systems in Scope:

The following systems will be used during testing:

System	Description
Horizon Core Applications	Reference Data Services Near Real Time Services Horizon Product Applications Audit Services Reconciliation Applications
Horizon Data Repositories	Branch database

Testing Scope:

Business Process Area	Level 3 Process	Business Process Area
Procurement	Requisitions	Requisitions creation
		Requisitions approvals
Finance	General Ledgers	G/L Journals
		G/L Reversals
		G/L Write-offs

3.5 Out of Scope

Any changes or testing relevant to the programme / project that is NOT covered here, and why (for example, Penetration Testing may be managed and delivered by an independent third party, reporting directly to the PM rather than Test Manager)

Example: negative testing



Programme Test Strategy v<x>



4 Cross-Project Organisational Test Strategy

The Organisational Test Strategy includes information on the standard testing methodology, approach, and processes that are applied across all projects. This Programme / Project Test Strategy aligns to that document, and so this information is not repeated here.

The following variations from the Organisational Test Strategy are noted here: (delete this if there are no variations)

Before completing this section, ensure you have referred to the Organisational Test Strategy (particularly Section 4 in the Organisational Test Strategy template) and that you are clear on whether this strategy aligns to it, and whether there are any variations.

List the variations, be clear on how the approach will vary, why variance is required / justified; and any associated risks and mitigations (which should also be in the RAID section #section ref TBC#



5 Programme / Project Test Strategy

This section supplements the Cross Project Organisational Test Strategy above and covers the specifics for the current programme or project. This is a key section in this document and should inform all stakeholders about the testing that will take place on this programme or project.

5.1 Programme / Project Risk Management

How risks will be managed, on an ongoing basis. Include information on how risks are escalated, and what governance forums are used in the organisation (e.g. Project SteerCo, Operational Risk Committees, etc)

5.1.1 Testing as a Risk Management Activity

Refer to the key changes being introduced by the programme or project, the risks associated with those changes, and how testing will be focussed on those risks.

Implementing technology change introduces a risk that new or changed services may not deliver expected business benefits, and the change may negatively impact existing services, potentially impacting business operations, finances, or customer experience.

Testing is a Quality Control activity, which as part of a Quality Assurance approach, is employed as a risk management activity to manage or mitigate this risk. The risks associated with change are assessed, and should inform the approach and scope to testing: what is changing, where are the likely impacts, what can be tested in advance to find out more about those potential impacts? Testing can therefore reduce risk, or provide a more informed risk assessment which can be used when considering whether changes can should be implemented into the live environment. The testing for each change will be informed by the risks associated with that change.

5.1.2 Risk Based Testing

If a Risk Based Testing approach is not being followed, please state that.

Describe the programme approach to Risk Based Testing, as appropriate. Be clear that RBT is a structured approach, and should include workshops with business and technical stakeholders to qualify and rate Impact of Failure (defined by business and live services stakeholders) and Likelihood of Failure (defined by technical teams). Also be clear that RBT is not purely a way to cut down on testing costs or timescales, but is there to prioritise important tests to ensure that critical defects are found earlier.

Tests Prioritization:

The Test Risk Assessment will enable the prioritisation of the Business Processes, allowing the test team to spend most of the effort testing the most risky areas. We use both indicators to define a priority for the test scenarios from the Business Impact and Technical Risk.

Business Impact:

This indicator comes from the business 'know how' and is achieved by answering questions during the Validate phase against the reference processes such as how often a process is run, what the legal or financial implications of a process may be or how long could the business live without a particular process.



Programme Test Strategy v<x>

Volume/Frequency ¹	Legal or Financial Impact	Impact on Business Continuity
1-High	1-High	1-Immediate
2-Medium	2-Medium	2-Days (1-5 days)
3-Low	3-Low	3-Weeks (> 5 days)
4-V Low	4-V Low	4-Monthly

¹ The scale for Volume/Frequency needs to be factored based on the future business e.g. for a global business this may be 1,000s a day, 100s a day, daily, > daily whereas for a low volume business this may be daily, weekly, monthly, yearly

Technical Risk:

The risk of failure will be defined with the functional team and/or solution designer from the project team. This risk relates to the complexity of the solution supporting the test scenarios i.e. complexity of system configuration and any custom code supporting the scenario (i.e. interfaces, reports, transactions, etc.) and the complexity of the system integration test scenarios itself.

Complexity of System/Level of Customisation	Complexity of scenario
1-High	1-High
2-Medium	2-Medium
3-Low	3-Low

<Text>

5.1.3 Management of Programme / Project Testing Risks

Describe how the testing function will raise / manage risks during the programme / project (for example, risk that the test environment may be delivered late) – usually by participating in RAID reviews and contributing to the project RAID log.

5.2 Programme / Project Test Approach / Methodology

State / summarise the approach and model being followed, If a non-standard approach is being followed, explain why the non-standard approach is being recommended.

This section should also describe the methodology applied, i.e. how different test approaches and methods are selected, according to the type of project or risks associated with the change (e.g. for low risk / high frequency agile website updates, an automated / risk based test approach may be applied; high risk / low frequency changes such as large-scale migrations may require a more bespoke test approach)

5.3 Testing Lifecycle

The testing lifecycle should align to the Test Policy. State / summarise the lifecycle being followed.

If a non-standard lifecycle is being followed, explain why the non-standard approach is being recommended.

Describe the testing lifecycle process which has specific steps to be executed in a definite sequence to ensure that the quality goals have been met. In STLC process, each activity is carried out in a planned and systematic way. Each phase has different goals and deliverables.

An example testing lifecycle is shown below:

- *Engagement – the test team are engaged on a project*
- *Estimation – the test team provide information on effort / duration of all testing activities*
- *Planning – the test team plan the testing activities, including production of a Project Test Strategy, Test Plans, team selection, etc*
- *Test Design and Preparation – the test team ramps up in size and activities, building the test execution schedule, writing test specifications, preparing test data, environments, etc, in readiness for test execution*
- *Test Execution – the test team execute tests, including defect management activities*

Completion – test execution finishes (according to agreed exit criteria) and the test team undertakes completion activities including writing test completion reports, updating regression packs, archiving

5.4 Testing Sub-Processes (Test Types / Phases)

Describe the different Test Types / Phases that are in scope for this programme / project, and include a diagram showing how the test types progress (for example a V-Model diagram, or an Agile iterative test cycle diagram).

Example:

Testing on the project will be carried out through the execution of defined Test Phases. The functional Test Phases should be sequential and transition from one phase to the next is governed by defined Entry and Exit Criteria. Each test phase builds on the previous test phase to collectively provide a comprehensive test approach with each phase targeting specific types of defects. The test phases control the variables, introducing more complexity gradually to make remediation easier and more efficient saving both time and resources.

Based on the scope of the project the following table outlines the standard test phases that are in-scope or the rationale for exclusion:

Test Phase	Description	In scope	Rationale for exclusion
Unit Testing	Formal testing by the development team of the completed developments to ensure they meet the requirements outlined in the specification document. Responsibility of the build team	Y	
System Testing	Formal testing that components built or changed meet the defined functionality within the constraints of the system and	Y	



Programme Test Strategy v<x>

Test Phase	Description	In scope	Rationale for exclusion
	<i>functional area. Responsibility of the Test Team</i>		
<i>SIT Testing</i>	<i>Formal testing of the end to end process flow including the integration points of a solution where the integration is either across functional areas within a single system or across multiple systems connected via interfaces. Responsibility of the Test Team</i>	Y	
<i>User Acceptance Testing</i>	<i>Formal and informal testing of business processes by Business representatives to give a user's perspective of the delivered solution to ensure the solution is suitable for the business. Responsibility of the Test Team to coordinate Post Office Business resources</i>	Y	
<i>Non-Functional Testing</i>	<i>Formal Testing of the performance and other non-functional requirements of the system for a new implementation through controlled testing via Performance Test tools and other methods. Responsibility of Post Office IT</i>	Y	
<i>Regression testing</i>	<i>Regression testing of a new update or patch release provided by the Vendor during a client implementation prior to applying the update/release to the project test environment. Responsibility of the Test Team</i>	Y	<i>Assumed that 21A vendor release takes place prior to UAT, so specific testing of 21A is not required. 21B will be deferred until after Go-Live.</i>

5.5 Test Selection and Prioritisation

Describe how tests are selected and prioritised, potentially used Risk Based Test approach.

Example: this point will be redirecting to point 4.1.2 from this document.

5.6 Testing degree of Independence

Detail out the level of independence from Development or 3rd Parties to show impartiality across the testing.



Example: if test team will be carried out by a third party, specific team, how will be the level of independent from the development team?

5.7 Governance

What specific governance is applied to testing on this programme / project, who approves documents and test readiness / completion, how are exceptions managed? Include some detail on who provides approvals at various stages during testing lifecycle, in particular approval of test plan, test readiness review, defect severity levels / closure and test completion.

Example: all the approvers will be on the Approval Authority from each document and also will be defined for the quality gates as part of the project roles and responsibilities.

5.8 Defect management

The key point of this section is to describe the approach to defect management (tooling, workflow, definitions, etc) that will be applied on this programme / projects, if different from the standard approach described in the Organisational Test Strategy.

Describe the Defect Management Process, including:

- *Triage Process*
- *Tools to be used for tracking and managing Defects*
- *Might be useful to add in a process flow for this from Visio or another tool.*
- *Example only do not use*

Example: This section describes the process by which defects will be captured and monitored during their lifecycle, from creation to closure.

A defect is defined as any failure that prevents testing being undertaken or completed. Defects will be raised during testing for any error that occurs that impacts the ability to test the system, business process and/or transaction.

All defects will be logged in the Programme Defect Management Tool where they will be reviewed and tracked until they are resolved.

A defect is considered resolved when it is in one of the statuses described below:

“Closed – Rejected” – It has been demonstrated that the process or system is operating as designed and agreed this with the test analyst(s) who reported the defect.

“Closed” – A fix has been applied and where necessary the change has been promoted to the test environment, and the tester has demonstrated that the process or system is now operating as designed.

“Change Request” – This status is set once there is agreement that the defect relates to a new requirement rather than the solution working as designed

Consider what and how Severities are going to be applied detail out what each of the rating will mean an example of this is detailed below:

Severity	Rating	Description
Severity 1	Critical	The failure causes the system crash, unrecoverable data loss, security issue or performance issue. The testing cannot proceed without fixing the defect.
Severity 2	High	Critical functions of the system are not available to all users or all functions are not available to some users. Workarounds are not available.
Severity 3	Medium	Some users unable to gain access to non-critical parts of the system. Workarounds are available.
Severity 4	Low	A cosmetic problem or one affecting functional areas of low criticality. Workarounds are available, or not required.

Consider what and how Severities are going to be applied detail out what each of the rating will mean an example of this is detailed below:

<i>Severity</i>	<i>Rating</i>	<i>Description</i>	<i>Guideline:</i> <i>Initial review and update defect</i>	<i>Guideline:</i> <i>Time to defect closure (indicative only)</i>
<i>Priority 1</i>	<i>Critical</i>	<i>The failure blocks or prevents progress on all testing (or a significant amount of testing), for example the system or environment is not available.</i>	<i>Immediate</i>	<i>1 day</i>



Programme Test Strategy v<x>

		<i>Priority 1 defects need to be resolved urgently, as soon as possible, as they block testing progress. An SLA may be agreed with the project team, for example 1 to 4 hours.</i>		
<i>Priority 2</i>	<i>High</i>	<i>The failure blocks an area of testing, for example a significant function which is required to run several tests, or an area of the system / infrastructure is not available. Priority 2 defects need to be resolved quickly, as they will impact the test schedule. An SLA may be agreed with the project team, for example 4 to 12 hours.</i>	<i>4 hours</i>	<i>1 - 2 days</i>
<i>Priority 3</i>	<i>Medium</i>	<i>The failure prevents or impedes some test execution, without impacting overall test progress.</i>	<i>1 day</i>	<i>3 - 5 days</i>
<i>Priority 4</i>	<i>Low</i>	<i>The failure does not prevent or impede any testing.</i>	<i>3 days</i>	<i>1 week</i>

5.8.3 Defect Management of Testing Completion

Describe what is to be done with the outstanding defects at the end of the testing phase concentrate on handing them over to Business as Usual or any subsequent phase which is outside of the remit of this Strategy.

Example: outstanding defects will need to be tested before go-live in a test environment

5.8.4 Defect Tooling

Refer to the tool(s) that are used to manage defects.

Example: JIRA with Zephyr Scale plug-in

5.9 Test Documentation and Deliverables

List the documents and deliverables, example list is shown below:

- *Test Plan (potentially one for each test type / phase)*
- *Test Specifications*
- *Test Progress Reports (include information on frequency, content, and audience)*
- *Test Completion Report*

5.10 Reporting and Metrics

Describe the reports and metrics that will be gathered, which relate to testing activities on the programme / projects.



Example: daily executions report, weekly testing activities report with the plan vs actual comparison and exit criteria measurements and other KPI's.

5.11 Configuration Management of Test Work Products

Describe the process for how documents will be issued, approved, and stored (for use during the project, and then properly archived to support future audits)

This needs to include detail on how changes to documents will be managed (e.g. some may just be minor updates documented in End of Test Report, some may be need docs to be re-issued and re-approved. A table showing levels of governance for different products and changes may help)



6 Testing Resources

6.1 Test Tools

This section is to briefly list the tools that will be used on this programme / project.

6.1.1 Test Management Tools

What test management tools are used, and what they are used for.

Example: JIRA with Zephyr Scale plug-in

6.1.2 Defect Management Tools

Describe what defect management tools are used, and how they are used in conjunction with the test management tool, and with external suppliers if appropriate.

Example: JIRA with Zephyr Scale plug-in

6.1.3 Test Execution Tools

Describe what tools are used to support manual test execution, e.g. test data creation / loading (e.g. Postman), data analysis tools (e.g. SQL), and so on.

Example: JIRA with Zephyr Scale plug-in for functional manual test executions

Test execution schedules will be defined for each test phase/cycle in the Phase Test Plans and set-up in Zephyr Scale. Test scripts will be executed as per the test phase schedule using the Zephyr Scale to record results and document test evidence. Testers will be asked to record the document numbers within the expected results section of the script to ensure that this document can be used for the remaining steps

6.1.4 Test Automation Tools

Describe what Test Automation Tools are used, and how they are used. May need information on how new automation tools are evaluated and integrated into any existing frameworks.

Example: Automation will be used whether is possible and is highly recommendable for regression testing. Due to the multiple systems and different complexities is very likely that multiple tools are required (Selenium, UFT, Cucumber, etc) as well as a very strong automation framework need to be in place.

6.1.5 Performance Testing Tool

Describe what Performance Test tools are used, and how they are used.

Example: LoadRunner



6.1.6 Other Test Tools

Describe any other tools that are used in testing, or required by the test team to perform testing activities. Optional Section.

Is possible that other tools are required for other testing purposes, for example: penetration testing.

6.2 Test Automation

Describe the programme / project approach to test automation, including information on frameworks and tools currently available. Also describe how the programme / project has decided whether to apply test automation, and any work that may be required to build new frameworks / install new tools / develop automated test cases / etc

6.3 Test Environments

State which test environment will be used. Refer to the Organisational Test Strategy, but add sufficient detail here to make it clear what environment will be used (including technical requirements, specifications, limitations, etc). If any new or different environments will be used, describe them and explain why they are being used.

What environments are available for use by projects, how are they maintained and supported, how are project environment requirements assessed and met. For roles, accountabilities and responsibilities see point 6.4.1.

6.4 Test Data

What test data is required for the programme / project, what is available, how is it managed and provisioned, how are project test data requirements assessed and met. Also make note of risks and controls that are in place to ensure compliance to GDPR and other regulations. Focus on test data that will actually be used on this programme / project as well as responsible and accountable teams for this. For roles, accountabilities and responsibilities see point 6.4.1.

6.5 Testing Organisation

Describe the testing team on the programme / project, i.e. the structure of the test team (including internal and external resources), and how the team is being resourced.

6.5.1 Key Roles, Accountabilities and Responsibilities

List the key testing roles, accountabilities and responsibilities that are specific to this programme / project. A table showing example testing (and test related) roles is shown below:

Role	Description
Head of Test	<ul style="list-style-type: none">• Manage and lead the test team• Owns the Test Strategy• Ensures that tools and resources are available to deliver testing• Etc
Test Manager	<ul style="list-style-type: none">• Produce Project Test Strategy



Programme Test Strategy v<x>

	<ul style="list-style-type: none"> • <i>Manage test preparation and execution activities</i> • <i>Etc</i>
<i>Test Analyst</i>	<ul style="list-style-type: none"> • <i>Analysis of requirements</i> • <i>Produce test specifications</i> • <i>Execute tests</i> • <i>Raise defects</i> • <i>etc</i>
<i>Defect Manager</i>	<ul style="list-style-type: none"> • <i>Provide and support defect management process and tools</i> • <i>Chair defect triage and defect review meetings</i> • <i>Ensure resolution of defects</i> • <i>etc</i>
<i>Test Environments Manager</i>	<ul style="list-style-type: none"> • <i>Responsible for provision and maintenance of test environments</i> • <i>Ensure that environment issues are resolved</i> • <i>etc</i>
<i>Test Data Manager</i>	<ul style="list-style-type: none"> • <i>Responsible for adherence to GDPR and other regulations</i> • <i>Provision of test data</i> • <i>Etc</i>
<i>Development Team Manager</i>	<ul style="list-style-type: none"> • <i>Application support during test execution, including defect resolution</i> • <i>Etc</i>
<i>Business Analyst</i>	<ul style="list-style-type: none"> • <i>Support during test prep activities (including reviews of test specifications)</i> • <i>Support for defect management process (including confirmation of business impacts)</i> • <i>Etc</i>

6.5.2 Test Team Structure

Add Team Structure maybe include a programme org Chart and a Team Org Chart.

6.5.3 Staffing and Training Needs

Are any new skills required for testing on this programme / project? Describe what the needs are, and how they will be met (also ensure that these are reflected in the RAID section #TBC#

Add knowledge and experience desirable for each area. Specific training needs for each project are not covered her, but this should include information on how skills will be developed and maintained, and how new skill requirements will be assessed. Describe how testing resources are managed and obtained (where additional resources are required).

7 Test Sub-Process (Test Type / Phase) Specific Information

The following section includes information on each testing sub-process (test type or test phase) that will be applied on the programme / project.

The different Test Types / Phases are described in the Organisational Test Strategy, so select the relevant test types from the text below and delete as appropriate.

7.1 Unit Testing

Definition	<i>Either the accepted definition within the organisation, or the ISTQB definition</i>
Summary	<p><i>Developers run Unit Tests on discrete components, to ensure that the component has been built successfully and can run without errors or failures. Components are tested in isolation, typically using stubs or drivers (rather than other systems or components) and manufactured test data.</i></p> <p><i>'White Box' test techniques are applied to check behaviour of code or hardware components, independent from overall system functionality.</i></p> <p><i>Unit testing is limited as it can check for basic build, coding, runtime, or compile errors; it does not validate that the component will be meet functional or non-functional system / user requirements.</i></p>
Purpose	<i>Low level functional test phase, to verify that discrete modules are built without basic errors</i>
Entry Criteria	<ul style="list-style-type: none"> <i>Code is compiled and can be executed</i>
Exit Criteria	<ul style="list-style-type: none"> <i>Tests are executed without any critical errors</i>
Completion / Acceptance Criteria	<i>Unit testing does not have a formal set of completion or acceptance criteria, other than that each code component can be executed without any critical errors which would prevent execution of System Tests</i>
Documentation and Reporting	<i>Developers produce informal report stating what was tested and what defects were raised / resolved</i>
Degree of Independence	<i>Low – typically executed by the developers who wrote the code</i>
Test Design Techniques	<i>White Box techniques, e.g. branch / statement coverage</i>
Traceability	<i>N/A</i>
Test Environment	<i>Development environment</i>
Metrics	<i>N/A</i>
Retesting and Regression Testing	<ul style="list-style-type: none"> <i>Failed tests will be re-executed following any changes to resolve errors.</i> <i>Regression testing is N/A</i>



7.2 System Testing

Definition	<i>Either the accepted definition within the organisation, or the ISTQB definition</i>
Summary	<p><i>Testers run System Tests on complete systems, to ensure that the overall system behaviour meets functional system / user requirements.</i></p> <p><i>'Black Box' test techniques are applied to check system behaviour at functional level, independent to how the code may be operating.</i></p> <p><i>System testing is limited as it will validate an individual system, but not end-to-end processes or data flows that go across different systems.</i></p>
Purpose	<i>Aims to test the functionality, operability, and robustness of discrete systems prior to SIT and UAT.</i>
Entry Criteria	<ul style="list-style-type: none"> <i>Unit Testing is complete without any critical errors / defects</i> <i>Confirmation of build to be deployed</i> <i>Scope and requirements baselined</i> <i>Test Plan issued</i> <i>Test environment built and accessible, ready for use in test execution</i> <i>Test data loaded and available for use in test execution</i> <i>Required test tools available and ready for use</i> <i>Testing resources available and ready for test execution</i> <i>Confirmation that build has been deployed</i> <i>Test available to be executed</i> <i>Support resources available</i>
Exit Criteria	<ul style="list-style-type: none"> <i>Any non-Passed tests documented and approved</i> <i>No outstanding Severity 1 or 2 defects</i> <i>Outstanding defects documented and approved</i> <i>All defects are documented</i> <i>De-scoped tests are documented and approved</i> <i>Any exceptions from agreed Test Plan are documented and approved</i> <i>Test Completion Report issued</i>
Completion / Acceptance Criteria	<i>New or changed system functions have been demonstrated to operate as expected, without any critical defects.</i>
Documentation and Reporting	<ul style="list-style-type: none"> <i>System Test Plan</i> <i>Test Progress Reports (during preparation and execution)</i> <i>Test Completion Report</i>
Degree of Independence	<i>Testing carried out by test team who are independent from the development team, but potentially part of the same organisation.</i>
Test Design Techniques	<i>Black Box test techniques, e.g. boundary value analysis</i>
Traceability	<i>Technical design specifications</i>
Test Environment	<i>Independent test environment, which contains a working version of the system under test, with appropriate test data available</i>



Programme Test Strategy v<x>

Metrics	<ul style="list-style-type: none"> • <i>Environment availability</i> • <i>Planned vs actual test execution</i> • <i>Defect data</i> • <i>Risks / Assumptions / Issues / Dependencies</i>
Retesting and Regression Testing	<ul style="list-style-type: none"> • <i>Failed tests will be re-executed following any changes to resolve errors.</i> • <i>Regression testing should take place, to ensure that the existing system functions are not negatively impacted by new / changed system functions</i>

7.3 System Integration Testing (SIT)

Definition	<i>Either the accepted definition within the organisation, or the ISTQB definition</i>
Summary	<p><i>Testers run SIT on complete systems, to ensure that end-to-end scenarios meet functional system / user requirements, and to ensure that data is successfully passed across systems.</i></p> <p><i>SIT covers testing of interfaces between discrete systems (to verify that data can be passed successfully) and the integration of discrete systems (to validate that end-to-end processes, which run through different systems, can be completed successfully).</i></p> <p><i>SIT includes the following types of test:</i></p> <ul style="list-style-type: none"> • <i>Verification of interfaces</i> • <i>Validation of data across interfaces</i> • <i>Error handling (failed interfaces or data errors across system interfaces)</i> • <i>Integration of discrete modules or systems</i> • <i>Validation of transactions processed across multiple systems</i> • <i>Verification that data models across systems</i>
Purpose	<i>Aims to test the functionality, operability, and robustness of integrated systems, and the interfaces between systems</i>
Entry Criteria	<ul style="list-style-type: none"> • <i>Exit criteria from previous test phases are met (or exceptions agreed)</i> • <i>Confirmation of build to be deployed</i> • <i>Scope and requirements baselined</i> • <i>Test Plan issued</i> • <i>Test environment built and accessible, ready for use in test execution</i> • <i>Test data loaded and available for use in test execution</i> • <i>Required test tools available and ready for use</i> • <i>Testing resources available and ready for test execution</i> • <i>Confirmation that build has been deployed (including non-functional elements)</i> • <i>Post deployment readiness checks completed</i> • <i>Test available to be executed</i> • <i>Support resources available</i>
Exit Criteria	<ul style="list-style-type: none"> • <i>Any outstanding Entry Criteria from previous phase are closed</i> • <i>Any non-Passed tests documented and approved</i> • <i>No outstanding Severity 1 or 2 defects</i> • <i>Outstanding defects documented and approved</i> • <i>All defects are documented</i> • <i>De-scoped tests are documented and approved</i> • <i>Any exceptions from agreed Test Plan are documented and approved</i> • <i>Test Completion Report issued</i> • <i>Stakeholders confirm that acceptance criteria are met</i>



Programme Test Strategy v<x>

Completion / Acceptance Criteria	<i>New or changed system interfaces and functions have been demonstrated to operate as expected (across systems), without any critical defects.</i>
Documentation and Reporting	<ul style="list-style-type: none"> • <i>System Integration Test Plan</i> • <i>Test Progress Reports (during preparation and execution)</i> • <i>Test Completion Report</i>
Degree of Independence	<i>Testing carried out by test team who are independent from the development team, but potentially part of the same organisation.</i>
Test Design Techniques	<i>Black Box test techniques, e.g. boundary value analysis</i>
Traceability	<i>Technical design specifications, Code of Connection (CoCo) documents, system interface specifications, data model specifications</i>
Test Environment	<i>Independent test environment, which contains a working version of the systems and interfaces under test, with appropriate test data available</i>
Metrics	<ul style="list-style-type: none"> • <i>Environment availability</i> • <i>Planned vs actual test execution</i> • <i>Defect data</i> • <i>Risks / Assumptions / Issues / Dependencies</i>
Retesting and Regression Testing	<ul style="list-style-type: none"> • <i>Failed tests will be re-executed following any changes to resolve errors.</i> • <i>Regression testing should take place, to ensure that the existing system functions are not negatively impacted by new / changed system functions</i>

7.4 User Acceptance Testing (UAT)

Definition	<i>Either the accepted definition within the organisation, or the ISTQB definition</i>
Summary	<i>Complete business processes (including end-to-end processes through different systems) are tested, in a live-like environment, using live-like data, in a manner that reflects the way the systems will be used by live users. UAT scoping, test case generation, and execution are all supported by business users (typically the business areas will work with the central UAT team to agree and complete these activities).</i>
Purpose	<i>Validation of end user / customer journeys or business processes</i>
Entry Criteria	<ul style="list-style-type: none"> • <i>Exit criteria from previous test phases are met (or exceptions agreed)</i> • <i>Confirmation of build to be deployed</i> • <i>Scope and requirements baselined</i> • <i>Test Plan issued</i> • <i>Test environment built and accessible, ready for use in test execution</i> • <i>Test data loaded and available for use in test execution</i> • <i>Required test tools available and ready for use</i> • <i>Testing resources available and ready for test execution</i> • <i>Confirmation that build has been deployed</i> • <i>Post deployment readiness checks completed</i> • <i>Test available to be executed</i> • <i>Support resources available</i> • <i>Business users available to support testing</i>



Programme Test Strategy v<x>

Exit Criteria	<ul style="list-style-type: none"> Any outstanding Entry Criteria from previous phase are closed Any non-Passed tests documented and approved No outstanding Severity 1 or 2 defects Outstanding defects documented and approved All defects are documented De-scoped tests are documented and approved Any exceptions from agreed Test Plan are documented and approved Test Completion Report issued Business users confirm acceptance of unresolved defects Stakeholders confirm that acceptance criteria are met
Completion / Acceptance Criteria	New or changed systems and functions have been demonstrated to operate as expected and are fit for purpose (from a business or user perspective); that end-to-end business processes can be completed successfully.
Documentation and Reporting	<ul style="list-style-type: none"> User Acceptance Test Plan Test Progress Reports (during preparation and execution) Test Completion Report
Degree of Independence	Testing carried out by business or user resources, supported by a test team who are independent from the development team, potentially from a different organisation (i.e. the client organisation rather than the supplier organisation)
Test Design Techniques	Black Box test techniques, e.g. boundary value analysis
Traceability	Business requirements
Test Environment	Independent test environment, which contains a working version of the systems under test, with representative data
Metrics	<ul style="list-style-type: none"> Environment availability Planned vs actual test execution Defect data Risks / Assumptions / Issues / Dependencies
Retesting and Regression Testing	<ul style="list-style-type: none"> Failed tests will be re-executed following any changes to resolve errors. Regression testing should take place, to ensure that the existing system functions are not negatively impacted by new / changed system functions UAT may contain a formal regression phase, to ensure that end-to-end business processes can be completed successfully, regardless of the changes being tested

7.5 Business Acceptance Testing (BAT)

If applied, this is generally similar to UAT, perhaps differentiated by BAT being executed by a different team than the team that executed UAT. Please use UAT table and edit accordingly.

7.6 Non-Functional Testing (NFT)

This include Performance Testing and OAT, as sub-sections of this point. More details can be found on the NFT Test Strategy document.

<Text>

Definition	Either the accepted definition within the organisation, or the ISTQB definition
------------	---



Programme Test Strategy v<x>

Summary	<p><i>NFT assesses the capacity of the target infrastructure (hardware, middleware, and software) can support the IT change being implemented and run the live services required.</i></p> <p><i>It may include the following:</i></p> <ul style="list-style-type: none"> • <i>Usability</i> • <i>Compatibility</i> • <i>Portability</i> • <i>Accessibility</i>
Purpose	<i>NFT tests that non-functional requirements for IT changes have been met (and that existing non-functional requirements are not compromised by IT changes)</i>
Entry Criteria	<ul style="list-style-type: none"> • <i>Exit criteria from previous test phases are met (or exceptions agreed)</i> • <i>Confirmation of build to be deployed</i> • <i>Scope and requirements baselined</i> • <i>Test Plan issued</i> • <i>Test environment built and accessible, ready for use in test execution</i> • <i>Test data loaded and available for use in test execution</i> • <i>Required test tools available and ready for use</i> • <i>Testing resources available and ready for test execution</i> • <i>Confirmation that build has been deployed (including non-functional elements of build)</i> • <i>Post deployment readiness checks completed</i> • <i>Test available to be executed</i> • <i>Support resources available</i>
Exit Criteria	<ul style="list-style-type: none"> • <i>Any outstanding Entry Criteria from previous phase are closed</i> • <i>Any non-Passed tests documented and approved</i> • <i>No outstanding Severity 1 or 2 defects</i> • <i>Outstanding defects documented and approved</i> • <i>All defects are documented</i> • <i>De-scoped tests are documented and approved</i> • <i>Any exceptions from agreed Test Plan are documented and approved</i> • <i>Test Completion Report issued</i> • <i>Stakeholders confirm that acceptance criteria are met</i>
Completion / Acceptance Criteria	<i>New or changed systems and functions have been demonstrated to be able to operate according to expected non-functional requirements</i>
Documentation and Reporting	<ul style="list-style-type: none"> • <i>Non-Functional Test Plan</i> • <i>Test Progress Reports (during preparation and execution)</i> • <i>Test Completion Report</i>
Degree of Independence	<i>Testing carried out by a test team who are independent from the development team, potentially with support from specialist external or technical teams</i>
Test Design Techniques	<i>Non-Functional Test techniques (list out details according to test scope / approach)</i>
Traceability	<i>Non-Functional Requirements (NFRs)</i>
Test Environment	<i>Independent production-like environment</i>
Metrics	<ul style="list-style-type: none"> • <i>Environment availability</i> • <i>Planned vs actual test execution</i> • <i>Defect data</i>



Programme Test Strategy v<x>

Retesting and Regression Testing	<ul style="list-style-type: none"> • <i>Risks / Assumptions / Issues / Dependencies</i> • <i>Failed tests will be re-executed following any changes to resolve errors.</i> • <i>Regression testing should take place, to ensure that the existing system functions are not negatively impacted by new / changed system functions</i>
----------------------------------	---

7.6.1 Performance Testing

Performance testing is classified as part of NFT. More details can be found on the NFT Test Strategy document.

<Text>

Definition	<i>Either the accepted definition within the organisation, or the ISTQB definition</i>
Summary	<p><i>Non-functional testing, to assess how systems perform when put subjected to higher amounts of usage or data processing.</i></p> <p><i>May include the following types of performance testing:</i></p> <ul style="list-style-type: none"> • <i>Load testing (numbers of users accessing systems)</i> • <i>Volume testing (amount of data being processed by systems)</i> • <i>Soak testing (behaviour of systems when put under a relatively high load or volume over a sustained period)</i> • <i>Stress testing (behaviour of systems when put extremely high load or volume)</i>
Purpose	<i>Validates whether the new or changed systems can operate according to the required levels of performance.</i>
Entry Criteria	<ul style="list-style-type: none"> • <i>Exit criteria from previous test phases are met (or exceptions agreed)</i> • <i>Confirmation of build to be deployed</i> • <i>Scope and requirements baselined (NFRs, SLAs, volumetrics, baseline performance metrics, etc)</i> • <i>Test Plan issued</i> • <i>Test environment built and accessible, ready for use in test execution</i> • <i>Test data loaded and available for use in test execution (NB performance testing may require a large amount of test data / user accounts)</i> • <i>Required test tools available and ready for use (including performance test tools)</i> • <i>Testing resources available and ready for test execution</i> • <i>Confirmation that build has been deployed (including non-functional elements of build)</i> • <i>Post deployment readiness checks completed</i> • <i>Test available to be executed</i> • <i>Support resources available</i>
Exit Criteria	<ul style="list-style-type: none"> • <i>Any outstanding Entry Criteria from previous phase are closed</i> • <i>Any non-Passed tests documented and approved</i> • <i>No outstanding Severity 1 or 2 defects</i> • <i>Outstanding defects documented and approved</i> • <i>All defects are documented</i> • <i>De-scoped tests are documented and approved</i> • <i>Any exceptions from agreed Test Plan are documented and approved</i> • <i>Test Completion Report issued</i> • <i>Stakeholders confirm that acceptance criteria are met</i>
Completion / Acceptance Criteria	<i>New or changed systems and functions have been demonstrated to be able to operate according to expected performance requirements</i>



Programme Test Strategy v<x>

Documentation and Reporting	<ul style="list-style-type: none"> • <i>Performance Test Plan</i> • <i>Test Progress Reports (during preparation and execution)</i> • <i>Test Completion Report</i>
Degree of Independence	<i>Testing carried out by a test team who are independent from the development team, potentially with support from technical teams</i>
Test Design Techniques	<i>Performance test techniques</i>
Traceability	<i>Non-Functional requirements (NFRs), Service Level Agreements (SLAs), volumetrics, baseline performance metrics</i>
Test Environment	<i>Independent production-like environment (suitably sized to represent production infrastructure, and capable of handling large amounts of data / users / transactions)</i>
Metrics	<ul style="list-style-type: none"> • <i>Environment availability</i> • <i>Planned vs actual test execution</i> • <i>Defect data</i> • <i>Risks / Assumptions / Issues / Dependencies</i>
Retesting and Regression Testing	<ul style="list-style-type: none"> • <i>Failed tests will be re-executed following any changes to resolve errors.</i> • <i>Regression testing should take place, to ensure that the existing system functions are not negatively impacted by new / changed system functions</i>

7.6.2 Operational Acceptance Testing (OAT)

Operational Acceptance testing is classified as part of NFT, and Service Readiness Testing is a sub-section from OAT. More details can be found on the NFT Test Strategy document.

<Text>

Definition	<i>Either the accepted definition within the organisation, or the ISTQB definition</i>
Summary	<p><i>OAT aims to establish whether new or changed software systems (which have already undergone functional testing) can operate in the target production infrastructure, and be accepted into live service</i></p> <p><i>Scope of OAT may include the following:</i></p> <ul style="list-style-type: none"> • <i>Disaster Recovery</i> • <i>Resilience and Failover</i> • <i>Backup and Restore / Recovery</i> • <i>Monitoring and Alerting</i> • <i>Backout</i>
Purpose	<i>Non-functional elements relating operation of system on target infrastructure, including service readiness; to allow acceptance of IT change into live service</i>
Entry Criteria	<ul style="list-style-type: none"> • <i>Exit criteria from previous test phases are met (or exceptions agreed)</i> • <i>Confirmation of build to be deployed</i> • <i>Scope and requirements baselined</i> • <i>Test Plan issued</i> • <i>Test environment built and accessible, ready for use in test execution</i> • <i>Test data loaded and available for use in test execution</i> • <i>Required test tools available and ready for use</i> • <i>Testing resources available and ready for test execution</i>



Programme Test Strategy v<x>

	<ul style="list-style-type: none"> • Confirmation that build has been deployed (including non-functional elements of build) • Post deployment readiness checks completed • Test available to be executed • Support resources available
Exit Criteria	<ul style="list-style-type: none"> • Any outstanding Entry Criteria from previous phase are closed • Any non-Passed tests documented and approved • No outstanding Severity 1 or 2 defects • Outstanding defects documented and approved • All defects are documented • De-scoped tests are documented and approved • Any exceptions from agreed Test Plan are documented and approved • Test Completion Report issued • Stakeholders confirm that acceptance criteria are met
Completion / Acceptance Criteria	New or changed systems and functions have been demonstrated to be able to operate within service requirements; and can be supported and maintained according to service requirements
Documentation and Reporting	<ul style="list-style-type: none"> • Operational Acceptance Test Plan • Test Progress Reports (during preparation and execution) • Test Completion Report
Degree of Independence	Testing carried out by a test team who are independent from the development team, typically reporting in to the organisation's Service Management function (rather than the delivery project, or development team)
Test Design Techniques	Non-Functional Test techniques
Traceability	Non-Functional requirements, service requirements including SLAs
Test Environment	Independent production-like environment, potentially Pro-Production
Metrics	<ul style="list-style-type: none"> • Environment availability • Planned vs actual test execution • Defect data • Risks / Assumptions / Issues / Dependencies
Retesting and Regression Testing	<ul style="list-style-type: none"> • Failed tests will be re-executed following any changes to resolve errors. • Regression testing should take place, to ensure that the existing services are not negatively impacted by new / changed system functions

7.6.2.1 Service Readiness Testing (SRT)

This is part of OAT. More details can be found on the NFT Test Strategy document.

<Text>

Definition	<i>Either the accepted definition within the organisation, or the ISTQB definition</i>
Summary	<p><i>Service Readiness Testing aims to establish whether new or changed software systems (which have already undergone functional testing) is ready to be deployed into production, to go into live service.</i></p> <p><i>Scope of SRT may include the following:</i></p>



Programme Test Strategy v<x>

	<ul style="list-style-type: none"> • <i>Service Readiness – IT processes such as incident / problem management, change management, etc</i> • <i>Operational Readiness, including elements such system monitoring and reporting,, data retention, etc</i>
Purpose	<i>Non-functional elements relating operation of system on target infrastructure, including service readiness; to allow acceptance of IT change into live service</i>
Entry Criteria	<ul style="list-style-type: none"> • <i>Exit criteria from previous test phases are met (or exceptions agreed)</i> • <i>Confirmation of build to be deployed</i> • <i>Scope and requirements baselined</i> • <i>Test Plan issued</i> • <i>Test environment built and accessible, ready for use in test execution</i> • <i>Test data loaded and available for use in test execution</i> • <i>Required test tools available and ready for use</i> • <i>Testing resources available and ready for test execution</i> • <i>Confirmation that build has been deployed (including non-functional elements of build)</i> • <i>Post deployment readiness checks completed</i> • <i>Test available to be executed</i> • <i>Support resources available</i>
Exit Criteria	<ul style="list-style-type: none"> • <i>Any outstanding Entry Criteria from previous phase are closed</i> • <i>Any non-Passed tests documented and approved</i> • <i>No outstanding Severity 1 or 2 defects</i> • <i>Outstanding defects documented and approved</i> • <i>All defects are documented</i> • <i>De-scoped tests are documented and approved</i> • <i>Any exceptions from agreed Test Plan are documented and approved</i> • <i>Test Completion Report issued</i> • <i>Stakeholders confirm that acceptance criteria are met</i>
Completion / Acceptance Criteria	<i>New or changed systems and functions have been demonstrated to be able to operate within service requirements; and can be supported and maintained according to service requirements</i>
Documentation and Reporting	<ul style="list-style-type: none"> • <i>Service Readiness Test Plan</i> • <i>Test Progress Reports (during preparation and execution)</i> • <i>Test Completion Report</i>
Degree of Independence	<i>Testing carried out by a test team who are independent from the development team, typically reporting in to the organisation's Service Management function (rather than the delivery project, or development team)</i>
Test Design Techniques	<i>Non-Functional Test techniques</i>
Traceability	<i>Non-Functional requirements, service requirements including SLAs</i>
Test Environment	<i>Independent production-like environment, potentially Pro-Production</i>
Metrics	<ul style="list-style-type: none"> • <i>Environment availability</i> • <i>Planned vs actual test execution</i> • <i>Defect data</i> • <i>Risks / Assumptions / Issues / Dependencies</i>



Retesting and Regression Testing	<ul style="list-style-type: none">• Failed tests will be re-executed following any changes to resolve errors.• Regression testing should take place, to ensure that the existing services are not negatively impacted by new / changed system functions
----------------------------------	--

7.7 Security Testing

This is a specialist test type, and the organisation may take the approach of using external suppliers to execute security testing (which includes Vulnerability Assessments and Penetration Testing). Please add information as appropriate. More details can be found on the NFT Test Strategy document.

7.8 Regression Testing

Regression testing should take place within each test type / phase, but if the organisation runs a distinct Regression Test phase, please include details here. Also, it is recommended that regressions packs are automated, see point 6.1.

7.9 Post-Implementation Testing

These activities may not be applied as managed test types / phases, so may be out of scope for this document; but if appropriate, please provide information here. Examples types of post-implementation testing are:

- *Checkout Testing – confirmation that changes have been deployed successfully (into production), and that the new / changed systems are operating as expected*
- *Beta – a new or changed system may run for a period in production with limited users or data, to ensure that all operations are functioning as expected*
- *Trial / Pilot – business units may wish to operate new systems in production, but with limited access to users / customers, in order to assess whether the business processes are operating effectively*



8 Risks, Assumptions, Issues and Dependencies (RAID)

This section is to list the specific RAIDs associated with testing on this programme / project – avoid adding generic RAIDs (such as “there may be delays to the start of testing”) unless they are noteworthy for this particular programme / project (e.g. “there may be delays to the start of testing as we know that the new environments have not been built yet”), and avoid adding all programme / project RAIDs which are not related to testing)

The following specific RAIDs have been identified on the programme / project:

Type	Item	Impact Description	Impact (H/M/L)	Likelihood (H/M/L)	Proximity	Mitigation / Resolution	Owner
R	There is a risk that...						
A							
I				N/A	N/A		
D							



9 Glossary