| | |
|---|---|
| **Document Title:** | HNG-X Testing Strategy |
| **Document Type:** | Strategy |
| **Release:** | Whole HNG-X Programme |
| **Abstract:** | This is a joint Fujitsu Services/Post Office document describing the strategic approach to be applied for all testing and integration activities performed for the HNG-X Programme. |
| **Document Status:** | APPROVED |
| **Originator & Dept:** | John Hunt (NOJ Ltd.) for HNG-X Programme |
| **Contributors:** | Chris Young, Lee Farman, Peter J Robinson, Dave Johns, James Stinchcombe, Peter Jobson, Niall Finnegan, Mark Taylor, Giacomo Piccinelli, Bill Reynolds |
| **Editor's Note:** | Version/Date in header and lists of approval authorities/reviewers updated prior to switching on Tracked Changes (to avoid adverse impact on pagination) |
| **Internal Distribution:** | *(For Originator to distribute following approval)*<br>**FUJITSU SERVICES LIBRARY** |
| **External Distribution:** | *(For Document Management to distribute following approval)*<br>**APPROVAL AUTHORITIES (as below)** |

***Approval Authorities:*** *(See CM/ION/078 for Approval roles)*[1]

| Name | Role | Signature | Date |
|---|---|---|---|
| Bill Reynolds | Fujitsu HNG-x Programme Manager | | |
| Mark Burley | Programme Manager (Post Office Ltd.) | | |
| Beverley Dunn | Head of Delivery (Post Office Ltd.) | | |

[1] CM/ION/078 is being revised to cover HNG-x.

# 0.0 Document Control

## 0.1 Document History

| Version No. | Date | Reason for Issue | Associated CP/PEAK/PPRR Reference |
|---|---|---|---|
| 0.1 | n/a | Not issued – working copy only | |
| 0.2 | 10th Feb 2006 | First Issue for initial review | |
| 0.3 | n/a | Not issued – working copy only | |
| 0.4 | n/a | Not issued – working copy only | |
| 0.5 | 27th Feb 2006 | Issued for wider review | |
| 0.6 | 10th Mar 2006 | Not issued – working copy only | |
| 0.7 | 17th Mar 2006 | Issued for final review and approval | |
| 1.0 | 21st Mar 2006 | Issued for approval. | |

## 0.2 Review Details *(see CM/ION/078 for Review roles)*

| | |
|---|---|
| Review Comments by : | As advised in calling note for review meeting |
| Review Comments to : | Originator & Document Management |
| *Mandatory Review* | |
| **Approval Authorities (as per cover page) plus:** | |
| *Role* | *Name* |
| Fujitsu HNG-x Chief Technical Officer | Giacomo Picconelli* |
| Fujitsu HNG-x Test Design Strategy | Peter J Robinson* |
| Fujitsu HNG-x Programmes Office Manager | Graham Chatten* |
| Fujitsu HNG-x Design Team Manager | Tom Northcott |
| Testing Manager (Post Office Ltd.) | Chris Young* |
| | |
| *Optional Review* | |
| *Role* | *Name* |
| Fujitsu HNG-x Performance & Migration Architect | James Stinchecombe |
| Fujitsu HNG-x Testing | George Zolkiewka |
| Fujitsu HNG-x Testing | Gaynor Simpson |
| Fujitsu HNG-x Application ArchitectLegacy Application Specialist | Dave Johns |
| Fujitsu HNG-x Development Manager | Mark Taylor |
| Fujitsu HNG-x Systems Management Architect | Glenn Stephens |

| | |
|---|---|
| Fujitsu HNG-x Service Manager | Peter Burden |
| Fujitsu HNG-x Infrastructure Architecture Leader | Nial Finnegan |
| Fujitsu HNG-x Quality Manager | Jan Holmes |
| Chief Architect (Post Office Ltd..) | Clive Read |
| Design Authority (Post Office Ltd..) | David Gray |
| Implementation and Migration Manager (Post Office Ltd..) | Will Russell |
| Commercial Manager (Post Office Ltd..) | Simon Glynn |
| Testing Specialist (Post Office Ltd.) | Lee Farman |

( * ) = Reviewers that returned comments

## 0.3 Associated Documents

| X-Ref | Reference | Ver | Date | Title | Source |
|---|---|---|---|---|---|
| [1] | PA/TEM/001 | 11.0 | | Fujitsu Services Document Template | PVCS |
| [2] | VI/STR/064 | 1.0 | 15/08/2005 | Testing Approach for the Horizon System | PVCS |
| [3] | VI/STR/062 | 2.0 | 16/11/2005 | Fujitsu Services Testing Strategy for Horizon System | PVCS |
| [4] | RM/STR/005 | 0.3 | 21/06/2005 | Horizon NG – Programme Test Strategy | PVCS |
| [5] | n/a | n/a | n/a | HNG req 2.1 sect16.xls (compliance matrix for HNG requirements on testing) | email |
| [6] | RM/ARC/002 | 0.3 | 17/11/2005 | HNG-X Solution Architecture Outline | PVCS |
| [7] | n/a | n/a | n/a | Branch Migration V0-2.doc (migration working paper – transaction data) | email |
| [8] | RM/STR/016 | 0.1 | 15/08/2005 | Horizon Data Centre Migration Strategy (migration working paper – data centre) | PVCS |
| [9] | n/a | n/a | n/a | Process_Test_Execution_Main.pdf and Process_Test_Planningandprep_Main.pdf (corporate testing processes) | CafeVic |
| [10] | DE/STR/010 | 0.3 | | Strategy for Delivering T-Release Content | PVCS |
| [11] | VI/STR/086 | 0.1 | | T-Release Testing Strategy (Post S92 Testing Strategy) | PVCS |
| [12] | | 0.1 | 08/03/2006 | Establishing and Assuring the HNG-X User Interface | email |
| Details for following documents to be added as they become available | | | | | |

| | | | | | |
|---|---|---|---|---|---|
| | | | | Development strategy | |
| | | | | Business requirements | |
| | | | | Operational Requirements | |
| | | | | Acceptance Criteria | |
| | | | | APOC Report | |
| | | | | TPOC Report | |
| | | | | Performance Management Strategy | |
| | | | | Pilot Strategy | |
| | | | | Implementation Strategy | |

**Unless a specific version is referred to above, reference should be made to the current approved versions of the documents.**

**On each revision of this document the above list of associated documents should be considered. If they have changed since last used, then any such changes may be relevant to the strategic approach presented by this document. Where any such changes have been reflected in revising this document, the reference given for the document concerned must also be updated to indicate the correct version number.**

**N.B. Printed versions of this document are not under change control.**

## 0.4 Abbreviations/Definitions

| Abbreviation | Definition |
|---|---|
| | |
| ADSL | Broadband, always-on data connection via phone lines (landline) |
| API | Application Program Interface – a published/approved interface through which to access an application for a given purpose |
| APOC | Architectural Proof of Concept – brief exercise prior to main development activity to prototype/prove architectural principles, probe solution characteristics and trial technology combinations |
| BIT | Business Integration Test – a sub-stage of SV&I as used for Horizon testing |
| BSTP | Business Specified Test Point – a supplementary specification to Business Threads, to cover an area of particular importance to Post Office Ltd, but where it cannot readily be fitted into a Business Thread |
| BT | Business Thread – a (typically lengthy) test scenario contrived to link together many areas of particular importance to Post Office Ltd, such that as the tests for these areas are one, one feeds the next, gradually maturing the data along a chain of system events, and so also proving data flow integrity |
| CAST | Computer Aided Software Testing – term used to refer to the use of purpose |

| | |
|---|---|
| | built management and automation tools for software testing |
| CIT | Component Integration Test – component level verification and integration stage, against the design, where an individual component is linked with its direct neighbours |
| CM | Configuration Management – used interchangeably to refer to the system/toolset, the organisation, the processes, and the principles of managing the configurable items making up the solution |
| CMMI | Capability Maturity Model Integration – a quality management model seeking to promote continuous improvement and increasing maturity of process and supporting information/data |
| CR | Change Request |
| CT | Component Integration – component level verification stage, against the design, where an individual component is exhaustively tested in isolation |
| DC1 / DC2 | Data Centres 1 and 2 |
| D&D | Design & Development |
| DIT | Direct Integration Test – formal validation of an external system interfaces, performed for Horizon |
| DR | Disaster Recovery |
| E2E | End-to-End testing stage - performed for Horizon by Post Office Ltd |
| ELT | Extended Link Test – verification stage, linking units across platforms, performed for Horizon |
| EOD | End of Day |
| EPIDD | External Physical Interface Definition Document |
| GPRS | Mobile phone communications for data transmission, always-on, faster than GSM |
| GSM | The most common mobile phone communications |
| GUI | Graphical User Interface |
| HLD | High Level Design |
| HLTP | High Level Test Plan |
| HNG | Horizon New Generation |
| I/F | Interface |
| I/O | Input/Output (usually disk reading and writing traffic) |
| ISDN | Digital dialup-based connection (voice and data) via phone line (landline), using multiple channels, typically faster than PSTN but slower than ADSL |
| IT (general context) | Information Technology |
| IT (testing context) | Integrity Testing – stage of validation and integration, focussing on recovery and resilience areas, seeking to prove retained integrity of data and continuity |

| | |
|---|---|
| | of process (this abbreviation not to be encouraged as it is often misinterpreted, being over-used in the industry in general, for integration test, install test, etc.) |
| IV | Interface Validation – misnomer, verification stage, against the design to confirm the structure of an individual (usually internal) interface, performed for Horizon |
| JAVA | A common O-O programming language |
| J2EE | JAVA based application programming for the Enterprise (web enabled) world |
| LAN | Local Area Network |
| LFS | Logistics Feeder System – one of the business applications within Horizon |
| LLTS | Low Level Test Script |
| LST | Live System Test – the final stage of testing for Horizon just before Go-Live, using a highly Live-like environment managed by the Live operational support team |
| LT | Link Test – verification sub-stage (part of Unit Test) , against the design, linking individual modules together into Units, performed for Horizon |
| MT | Module Test – verification sub-stage (part of Unit Test), against the design, testing an individual module, performed for Horizon |
| NFR | Non-Functional Requirement |
| NT4 | Microsoft's NT version 4 operating System for the PC |
| O-O | Object-Oriented, or Object-Orientation |
| PAT | Product Acceptance Test – brief confirmation that third party deliverable meets minimum acceptable standard, so as to insulate later testing stages from potential disruption |
| PC | Personal Computer |
| PCT | Platform Configuration Test – verification stage, against the physical platform design and the application configuration design, performed for Horizon |
| PED | Physical Environment Description |
| PL/SQL | Common programming facility for manipulating data on relational databases such as provided by ORACLE |
| POA | The Post Office Account within Fulitsu Services |
| POL | Post Office Limited |
| POL-FS | Post Office Ltd's Financial Systems |
| PRF | Problem Review Forum – joint supplier/customer forum discussing and prioritising problem reports mostly arising from testing |
| PSTN | Standard (non-digital) dialup-based connection (voice and data) via phone line (landline), using a single channel, typically slower than ISDN and ADSL |

| | |
|---|---|
| PT | Performance Testing – loosely related group of testing activities confirming system performance, throughput, volumes, response time and behaviour under stress (this abbreviation to be avoided as it is ambiguous) |
| RAB | Release Authorisation Board |
| RAC | Real Application Cluster – an ORACLE technology for high performance implementations where multiple hardware platform servers need access to a common ORACLE database |
| RUP | Rational Unified Process – proprietary system development lifecycle methodology for iterative developments |
| RV | Release Validation – release level validation and integration stage, preparing the solution for release and confirming it is acceptable |
| SAN | Storage Area Network |
| SRDF | Proprietary high speed, high resilience, interlinking/synchronising connection between Sequent nodes (Host system platforms), typically at different geographical locations |
| ST | System Test – validation stage (though performed within the development arena), against the system requirements, integrating all the necessary component to form a discrete system within the overall solution |
| SUC | System Use Case |
| SUCM | System Use Case Model |
| SV&I | Solution Validation and Integration (or for Horizon, Sytem Validation and Integration) – solution level validation and integration stage, assembling all the systems together to form the whole solution and validating it against the requirements |
| TBC | To be completed – an area of the document to be completed at some future point when the necessary information emerges as the project lifecycle unfolds |
| TBD | To be determined – a particular point as yet unknown, yet to be determined in detail |
| TCO | Total Cost of Ownership |
| TED | Technical Environment Description |
| TPOC | Testing Proof of Concept – devising, developing and piloting any new or changed testing processes, tools and automation methods, and environment management and build procedures, in advance of having to use them in anger in the mainstream programme delivery |
| TPS | Transaction Processing System – one of the business applications within Horizon |
| UAT | User Acceptance Test – formal stage of testing performed by end users to confirm acceptability of a delivered system (no longer performed for Horizon, but a common concept within the IT industry) |
| UI | User Interface (see also GUI) |

| | |
|---|---|
| UML | Unified Modelling Language – standard language for requirement and system specification |
| UT | Unit Test – verification stage on Horizon which includes MT and LT |
| VT | Volume Test (see also PT) – (this abbreviation to be avoided as it is ambiguous) |
| WAN | Wide Area Network |
| XP | Microsoft's Windows XP operating system for PCs |

## 0.5 Changes in this Version

| Version | Changes |
|---|---|
| 0.1 | document in initial production |
| 0.2 | Prepared for first issue and review |
| 0.3 | Comments Applied |
| 0.4 | Populate main 'Approach' section |
| 0.5 | Prepared for wider review |
| 0.6 | Comments from review meeting applied, Abbreviations added, distribution/ reviewer/approval authority details corrected, initial assumptions/risks/ dependencies/constraints added |
| 0.7 | Prepared for final review and appoval |

## 0.6 Changes Expected

| Changes |
|---|
| |
| |
| |
| |
| |
| |
| Future issue(s): |
| Tailor list of principal test motivators, populate the section on Regression Testing & Retesting, expand the section on Progress by Achievement, and populate the sections on Outsourced Developments and In-House Developments, when the development strategy becomes available and the corresponding deliverable types are known |
| Revise to reflect the lessons learned from the Architecture Proof of Concept (APOC) and Testing Proof of |

| |
|---|
| Concept (TPOC) exercises, and the additional information emerging during the Systems Analysis and High Level Design stages of the Programme, and in particular more comprehensive entries for Environments Tools & Automation, and for Test Coverage |
| Expand sections on Assumptions, Dependencies, Risks, and Constraints as they develop |
| Populate section on Security Testing, when security requirements known and the access security policy and security specification become available |
| Populate sections on Environments, Tools & Automation and on Metrics, when the environment and automation strategy has been determined (comes out of the TPOC) |
| Populate sections on Features to be Tested and Features not to be Tested, initially at an outline level following prioritisation of the Test Objectives arising from analysis of the System Requirements, and later in more detail following completion of the high level test analysis activity, where details of the relevant features will emerge (prior to completion of the HLTPs) |
| Update section on Usability when approach for definition, assurance, and acceptance has been determined and agreed. |

## 0.7 Table of Contents

1.0 INTRODUCTION......3
1.1 BACKGROUND......3
1.2 PURPOSE OF DOCUMENT......3
1.3 SCOPE OF DOCUMENT......3
1.4 CONTEXT......3
1.5 COMPLIANCE......3
2.0 RATIONALE & OUTLINE APPROACH......3
2.1 CONSIDERATIONS......3
2.1.1 Existing Horizon Solution......3
2.1.2 Project Lifecycle Methodology, and Organisation......3
2.1.3 Requirements & Acceptance Criteria......3
2.1.4 Solution Architecture......3
2.1.4.1 Infrastructure......3
2.1.4.2 Counter Layer......3
2.1.4.3 Web Services Layer......3
2.1.4.4 Branch Database......3
2.1.4.5 Agent Layer......3
2.1.4.6 Host Layer......3
2.1.4.7 External Interface Layer......3
2.1.4.8 Disaster Recovery Provision......3
2.1.5 Data Centre Strategy......3
2.1.6 Release Structure......3
2.1.6.1 Outline Structure......3
2.1.6.2 Data Centre Migration......3
2.1.6.3 Branch (Counter) Migration......3
2.2 IMPLICATIONS FOR TESTING & INTEGRATION......3
2.2.1 No Reprieve … yet (Heavy Duty Testing)......3
2.2.2 Iterative Development & Integrated Project Teams......3
2.2.3 Collaborative Working & Merging Test Stages......3
2.2.4 Robust Component Level Verification becomes Essential......3
2.2.5 Non-Functional Aspects – Can't be Taken for Granted......3
2.2.5.1 Performance......3
2.2.5.2 Integrity (Resilience & Recovery)......3
2.2.5.3 Disaster Recovery......3
2.2.5.4 Operability and Systems Management......3
2.2.5.5 Security......3
2.2.5.6 Usability......3
2.2.6 Migration Complex and Critical......3
2.2.7 New/Revised Processes & Tools......3
2.3 APPROACH REQUIRED......3
3.0 GOALS & OBJECTIVES......3
3.1 TEST MISSION (TOP LEVEL TESTING GOALS)......3
3.2 TEST MOTIVATORS (KEY SOURCES DETERMINING TESTS)......3
3.3 HIGH LEVEL OBJECTIVES (THE PRINCIPLES)......3
4.0 STRATEGIC APPROACH......3
4.1 OBJECTIVE DRIVEN TESTING......3
4.1.1 Risk Based Foundation......3

4.1.2 Objective Driven Method 3
4.1.3 The Principles 3
4.1.4 Outline Process 3
4.2 FORMAL PLANNING & PREPARATION 3
4.3 PROGRESSIVE INTEGRATION 3
4.4 STAGES & TYPES OF TESTING & INTEGRATION 3
4.4.1 Existing Horizon Testing Stages 3
4.4.2 Addressing the Issues 3
4.4.2.1 Component Level 3
4.4.2.2 System level 3
4.4.2.3 Solution Level 3
4.4.2.4 Release Level 3
4.4.3 HNG-X Testing Stages 3
4.5 REGRESSION TESTING & RETESTING 3
4.6 FLEXIBLE IMPLEMENTATION 3
4.7 PROGRESS BY ACHIEVEMENT 3
4.8 HORIZON BASED AREAS 3
4.9 NEWLY DEVELOPED AREAS 3
4.9.1 Outsourced Developments 3
4.9.2 In-House Developments 3
4.9.3 Iterative Lifecycles and Integrated Mixed-Discipline Project Teams 3
4.10 PARALLEL TESTING 3
4.11 COLLABORATIVE WORKING 3
4.12 DEFECT MANAGEMENT 3
4.13 ENVIRONMENTS, TOOLS & AUTOMATION 3
4.14 METRICS 3
4.15 PROCESS IMPROVEMENT 3
**5.0 TESTING COVERAGE 3**
5.1 FEATURES TO BE TESTED 3
5.2 FEATURES NOT TO BE TESTED 3
**6.0 PEOPLE, SKILLS & RESPONSIBILITIES 3**
**7.0 ASSUMPTIONS, DEPENDENCIES, RISKS & CONSTRAINTS 3**
7.1 ASSUMPTIONS 3
7.2 DEPENDENCIES 3
7.3 RISKS 3
7.4 CONSTRAINTS 3

## 0.8 Table of Contents

Figure 1 : Scope of Testing Covered by this Document........................................................3
Figure 2: Document Context Map..............................................................................................3
Figure 3 : Outline Solution Architecture.................................................................................... 3
Figure 4 : Use of Branch Routers................................................................................................3
Figure 5 : Branch Database – Logical Sub-divisions.................................................................3
Figure 6 : Branch Database – Logical Partitioning (Chunking)................................................3
Figure 7 : DR Configurations – Active/Active vs Active/Standby............................................3
Figure 8 : Outline Release Structure...........................................................................................3
Figure 9 : Data Centre Transition through Hybrid Configuration.............................................3
Figure 10 : Migration of Branch Data......................................................................................... 3
Figure 11 : Objective Driven Testing - Outline..........................................................................3
Figure 12 : Outline Approach.......................................................................................................3
Figure 13 - Objective Driven Testing – Process Flow................................................................3
Figure 14 : Existing Horizon Testing Lifecycle...........................................................................3
Figure 15 : Component Test Context.......................................................................................... 3
Figure 16 : Component Integration Test Context....................................................................... 3
Figure 17 : Testing Stages – Overlap........................................................................................... 3
Figure 18 : HNG-X Testing Lifecycle Overview......................................................................... 3
Figure 20 : Fully Iterative Development Lifecycle...................................................................... 3
Figure 21 : Testing Lifecycle – Always Iterative......................................................................... 3
Figure 22 : Parallel Testing for HNG-X...................................................................................... 3

# 1.0 Introduction

## 1.1 Background

The Horizon system was initiated in 1995. It was initially intended both to rollout IT automation across the national network of Post Offices, and to implement a card based benefit payment system at those Post Offices, for the Benefits Agency, to help address concerns over benefit fraud.

Over the years it has undergone a series of very significant changes, in response to equally significant changes in business strategy, and also to exploit technological advances. However, whilst different applications have come and gone, and the supporting infrastructure has continued to evolve, the underlying solution architecture remains essentially the same.

It has been recognised for some time that this architecture, whilst providing an extremely robust operational solution, was not ideally suited to the very different business and technology drivers that prevail today. In addition, in common with many elderly systems that have been subjected to a succession of major changes, it has become increasingly difficult to make those changes, and expensive to operate.

In response to this, the Horizon New Generation (HNG) programme proposed to entirely re-engineer the solution, and to refresh the hardware across the national network of Post Offices. The main drivers were to create a solution that was more responsive to business change (faster time to market), and more efficient to operate, maintain, and enhance, thus providing lower Total Cost of Ownership (TCO). However, HNG was ambitious, projected costs were high, and the benefit realisation profile was unclear. In particular, HNG had been predicated on expecting a high rate of future business change for the system. Emerging business strategy in the Post Office indicated that this was uncertain, and could not be relied upon sufficiently to support the proposed business case. As a result HNG was suspended in the summer 2005.

The HNG-X programme proposes a somewhat less ambitious re-engineering of Horizon, without the branch network hardware refresh, and with the focus squarely on reduction of the TCO. The principal drivers for the HNG-X programme are to deliver a solution that significantly reduces the TCO, whilst maintaining ‘Business Equivalence’ (i.e. the HNG-X solution is to provide effectively the same business capability as the existing Horizon solution, but cost less to operate and maintain).

## 1.2 Purpose of Document

The primary purpose of this document is to set the strategic direction for the testing and integration effort required across the whole HNG-X programme.

Consistent with the general aim to reduce TCO, this is a joint document, meeting the needs of both Fujitsu Services and Post Office Limited, rather than each organisation producing separate documents. But further than this, the intention is to adopt a single consistent approach to the testing and integration for HNG-X, exploiting synergy wherever possible, to eliminate unnecessary duplication of effort and to reduce the overall elapsed time consumed without compromising product quality.

To this end it is intended to:

- Provide a central artefact defining the strategic approach with which to govern the testing and integration activities necessary for HNG-X, both for Fujitsu Services and for Post Office Limited.
- Provide visibility to the stakeholders concerned, that adequate consideration has been given to various aspects influencing the testing and integration required, and where appropriate to have those stakeholders approve the strategy.
- Communicate the agreed approach to all relevant parties, explaining how the approach addresses the principal drivers of the programme, defining at an outline level the scope and coverage of the testing required, and identifying the stages and types of assurance and testing to be employed.
- Indicate the approach to be adopted for tools, automation, environments, test management, defect management, metrics, traceability, and configuration management.
- Highlight the key risks, assumptions, dependencies, constraints, and responsibilities.

## 1.3 Scope of Document

This strategy covers all the testing and integration activities required across the whole HNG-X programme, for both Fujitsu Services and Post Office Limited. It spans from initial involvement in the Requirement Analysis stage, through Design and Build, up to and including pre-Live Operational Proving, and also covers the testing and integration of the XP release, following shortly after completion of the counter application migration.

It is concerned with all aspects of that testing and integration, encompassing both Static and Dynamic forms. It covers both verification and validation perspectives, and also the integration of software components, systems, and the whole solution.

However, this strategy is confined to covering only the testing and integration activities required for the HNG-X programme. It does consider the Horizon systems, but only in so far as the HNG-X solution involves re-engineering and/or re-deployment of those systems, and in so far as the two architectures coexist in hybrid form during the counter application migration period. (See [10] & [11] for relationship to Horizon T-Releases and the strategy for testing them.)

It is not concerned with the testing and integration required for the succession of Horizon releases continuing in parallel with and in preparation for the HNG-X programme – this will be covered in separate testing strategy documents specific to those Horizon releases.

Similarly, it is not concerned with the ongoing testing and integration of the HNG-X solution to take place for possible future releases following the completion of the HNG-X programme. Again, that will be covered in separate documents.

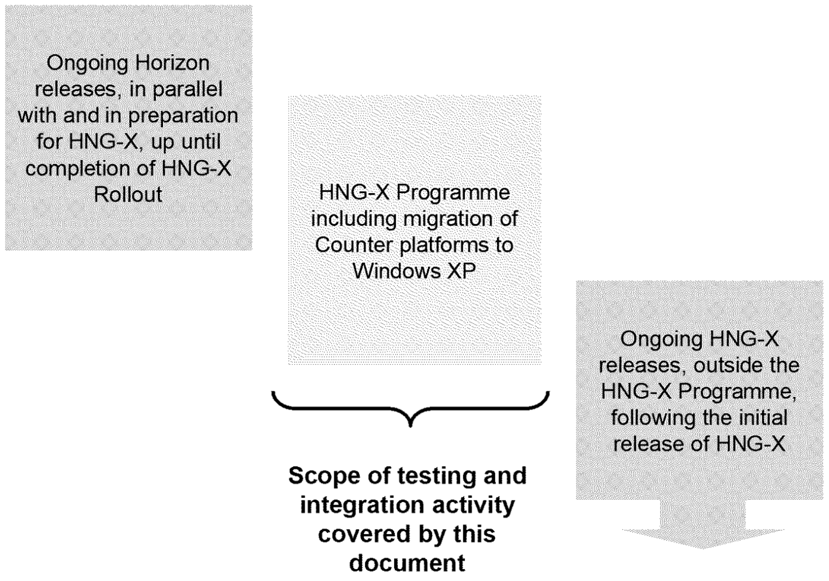

Ongoing Horizon releases, in parallel with and in preparation for HNG-X, up until completion of HNG-X Rollout
HNG-X Programme including migration of Counter platforms to Windows XP
Ongoing HNG-X releases, outside the HNG-X Programme, following the initial release of HNG-X
Scope of testing and integration activity covered by this document

**Figure 1 : Scope of Testing Covered by this Document**

## 1.4 Context

This document provides the high level strategic direction for all the testing and integration effort needed to meet the required scope (see 1.3 above). It includes an outline of the intended testing coverage, and the stages and types of testing necessary. It is specific to the HNG-X programme and stands independent of the strategies previously agreed for Horizon (see [2], [3], [4], [5], [10] & [11]). Nonetheless, these earlier documents remain relevant. In particular, to assist continuity, much of that earlier terminology has been retained, having become familiar through long use on Horizon. Similarly, corporate processes and CMMI requirements are also taken into account (see 1.5 below).

As this version of the document, produced during the Requirement Analysis stage of the programme, it will naturally be subject to the limitations imposed by the prevailing information. It is intended that it will be revised iteratively during the Systems Analysis and High Level Design stages, to reflect the emerging information about the systems under test and the lessons learned from the Testing Proof of Concept (TPOC) exercises. It should however remain at a high level, and it is not intended to expand this document to define the detailed testing coverage and methods required. Such material (aimed at a much narrower audience) should be defined separately in a series of subordinate documents, as shown in the following document context map.

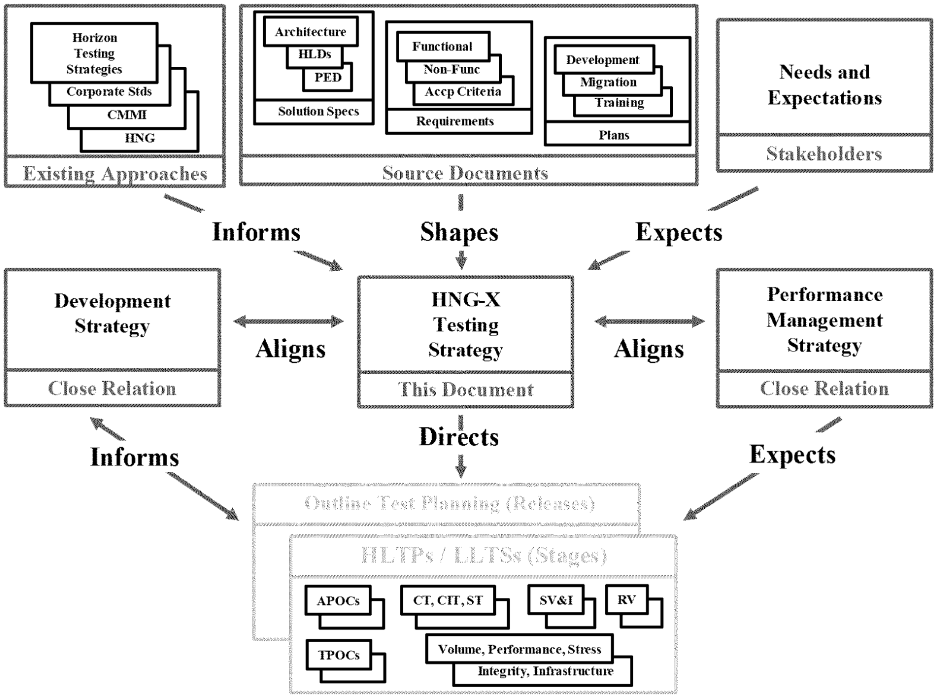


**Figure 2: Document Context Map**

As this document is intended for a diverse audience of stakeholders, test managers, testers, and other interested parties, it presents a number of different perspectives. The following guidance is provided to assist the reader in selecting the sections of the document likely to be most relevant.

All parties are encouraged to read the first three sections, which provide a brief introduction to HNG-X and to this document; an outline of the testing approach with the rationale for selecting it; and the high level objectives for testing.

Most readers are also likely to have an interest in sections 7, which highlights the key assumptions, dependencies, risks and constraints.

Stakeholders wishing just to confirm that the necessary considerations have been given and that an appropriate approach has been selected (but who do not need to follow the approach, or otherwise align their work streams with it) should find that section 2 provides a sufficient summary of the approach. In this case they may choose to omit sections 4, 5, and 6.

Stakeholders and other interested parties who do need to follow or in some way align their work streams with the approach are advised to include section 5 which gives a more comprehensive description of the approach and provides information on the different stages and types of testing employed, and section 6, which describes the various responsibilities concerned.

Any stakeholder or other interested party having a particular focus of interest (e.g. environments) should scan the table of contents for particular sections of interest (for this example, section 4.1.1 on environments).

Testers, Test Managers, and any other parties involved in following the approach, or who may be directly affected by it, should read the entire document.

## 1.5 Compliance

A number of compliance areas have been taken into consideration for this document, including existing Fujitsu Services Post Office Account (POA) processes, wider Fujitsu Services Corporate Processes, POA's adoption of CMMI, and the Post Office Limited (Post Office Ltd) initiative known as Harmony.

Overall it is the existing POA processes, as used for Horizon, which carry most relevance for this document. A strong preference has been expressed by Post Office Ltd (the Customer) to continue using the existing terminology where practicable. As this is a joint document, for both POA and Post Office Ltd, adopting a common and familiar terminology is very important. Also, as the Horizon systems feature heavily within the testing and integration of the HNG-X solution (see 2.1.1 below) then the investment made in these existing procedures has a significant bearing. Under the circumstances the wider Fujitsu Services Corporate Processes are not therefore considered appropriate here,and it is not intended to migrate to them for the testing of the HNG-X programme. The existing testing and integration processes, and the new facets of this HNG-X approach, are all entirely consistent with the intended adoption of CMMI, and care will be taken to ensure this continues. In this respect there has been particular attention to ensure the testing materials continue to be fully traceable back to the Requirements. At the time of writing, Post Office Ltd has indicated that the Harmony initiative does not appear to present any particular requirements for this document, but should any emerge as the initiative advances, then both parties will work together to accommodate them, under normal change control.

The existing terminology will therefore be retained for all testing and integration activities where this strategic approach is NOT calling for a significant change to established practice. Minor changes in approach will be reflected in correspondingly minor revisions to the existing processes. In this way both POA and Post Office Ltd will benefit from having built up a common understanding of the integration and testing domain. In the few areas where this approach is calling for a significant change in practice (such as adopting object-based component testing methods) then appropriate new terminology will be adopted to reinforce the need for a change in practice. Here common industry-wide terminology will be adopted (such as Component Testing, and Component Integration Testing). These changes (minor and major) will be developed and proven during a Testing Proof of Concept (TPOC) at the outset (see 2.3 below).

# 2.0 Rationale & Outline Approach

This section considers the HNG-X programme and the proposed HNG-X solution from a number of key perspectives, in order to identify what the special factors are about HNG-X as distinct from those of other projects. It examines the implications these special factors have for testing and integration, and describes in outline the strategic approach that these special factors demand. (As such, this section can be used in effect as a management summary of the approach, together with the rationale leading to it.)

Whenever at some point in the future a strategic change to the programme is considered, then from a testing perspective it is these special factors, and their consequential implications, that must be revisited in order to assess the likely impact on this strategic approach.

To help set the information given below in context, the following schematic illustrates the outline architecture proposed for HNG-X.

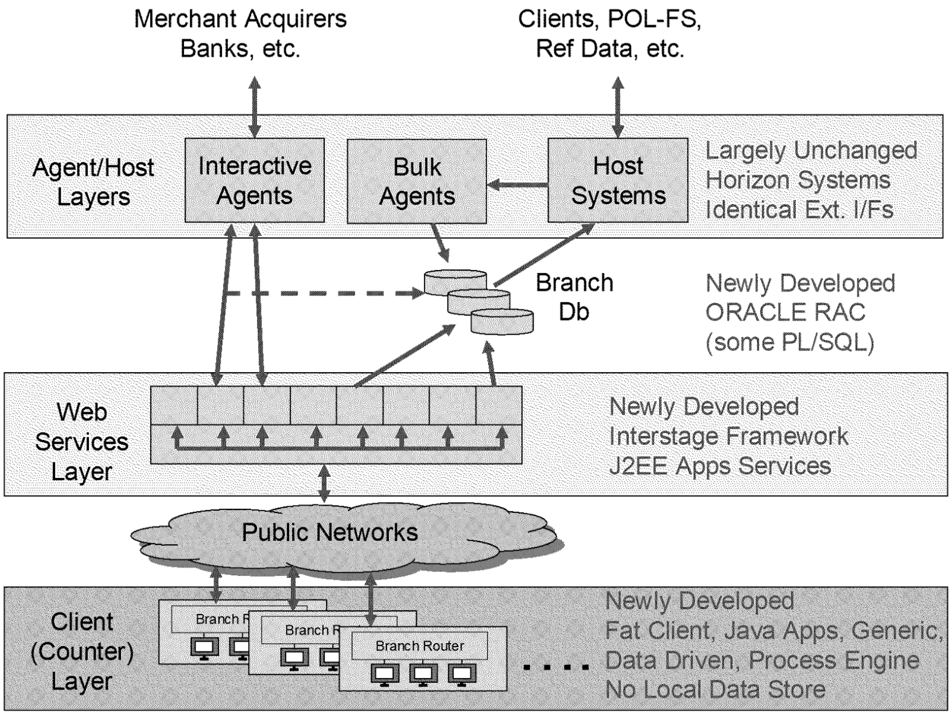


Figure 3 : Outline Solution Architecture

## 2.1 Considerations

There are a great many factors that can impact testing and integration. Some may make it more or less difficult to perform, or demand different levels of thoroughness.

Others may affect the relative priorities, or perhaps create inter-dependencies that force a particular sequence of activity. Each programme of work will have its own mix of such factors, which in turn determine the optimum strategic approach to adopt. Not surprisingly, some factors are more influential than others.

Here the most significant factors influencing testing and integration for the HNG-X programme are considered. They fit into 6 main categories, as follows:

### 2.1.1 Existing Horizon Solution

An important facet of the HNG-X programme is that it consists primarily of re-engineering / re-deployment of the existing Horizon solution. Significantly, from an application perspective, the boundaries of changes are clean.

- The Host systems, and their external interfaces, remain unchanged
- The Agent systems remain largely unchanged, with the impact mostly confined to the area of their current interface to the Correspondence Servers, which are changed to reflect their new interface with the Branch Database.
- The Correspondence Servers are entirely replaced by the new Branch Db.
- The Counter Applications and the associated Riposte Desktop and GUI, are entirely replaced by a new Client layer, developed in Java, and employing a data driven Process Engine.
- The Riposte Message Server (data replication) becomes obsolete, being replaced by the Client Server relationship to the Counter Applications, utilising a new Web Services framework to orchestrate the corresponding new Application Services at the centre. This also removes the need for a local Riposte Message Store at the counter, with all transaction history being retrieved from the centre in customary Client Server fashion.

The infrastructure changes necessary for HNG-X are less finely delineated. At the broad architectural level the major blocks of infrastructure systems required remain much the same as that for Horizon - Software Distribution, Systems Management, File Transfer Management, Asset Management, Security, Networks & Network Management, Resilience and Recovery, Disaster Recover, Peripheral Management, Time Synchronisation, End of Day Coordination, Batch Scheduling, etc. However, although the detail of it has not yet been determined, it is clear that below this high level view the supporting infrastructure must undergo significant and widespread revision, touching almost every area of the solution. Infrastructure should therefore be considered as a major area of change, requiring commensurate levels of testing.

It is also important to recognise that the existing Horizon systems will continue to operate in production, and continue to be subject to all necessary business driven enhancements, Live fixes, and other such changes, all in parallel with the development, testing, and deployment of the HNG-X solution.

Further, from the point when the HNG-X solution enters its deployment stage, through to the completion of its rollout, there will be a hybrid Horizon + HNG-X architecture in production, with the two solutions co-existing.

### 2.1.2 Project Lifecycle Methodology, and Organisation

In contrast with the previous intention for the HNG programme, no all-embracing move to RUP based development is planned for HNG-X. Appropriate iterative development techniques will be employed for those areas where it is deemed beneficial (e.g. the GUI), whilst existing development lifecycle structure (which is largely V-model based) will remain where there is no clear advantage in changing.

Also in contrast with HNG, where a substantial sized team was already established, HNG-X will need to build up the necessary resources from scratch. However, a significant proportion of the new development workload will be outsourced offshore, thus avoiding the need to build up a massive in-house capability. Such outsourced work will be actively progressed by the HNG-X programme, from within, in order to mitigate the risk of any potential communication/interpretation issues. (It is worth clarifying here that, from an integration and testing perspective, it is the fact that work is outsourced that needs to be considered, and not the fact that it may be going offshore. The key factor is that the necessary control processes must be in place for outsourced work to ensure that later stages of testing are not disrupted (i.e. the implicit assumptions built into the approach, about the rigour of the verification carried by the development area, must be confirmed for outsourced areas, to prevent potential disruption of the later stages of testing, which are dependent on the incoming deliverables having reached an appropriate level of readiness).

To provide the necessary contractual basis the Business Requirements and associated Acceptance Criteria (see 2.1.3 below), and the proposed Solution Architecture will all be developed in full at the outset (again in contrast with RUP style developments).

With HNG-X and the desire to reduce TCO comes a strong drive to remove duplication of effort, and to adopt collaborative working structures wherever this may assist. Indeed, this is already in evidence with the collaborative work in producing this joint strategy document. It is recognised that preparing for, conducting, and supporting an entirely separate stream of Customer Testing (by Post Office Ltd) may not be the most efficient organisation. There is a desire that this and any other potential opportunities be considered to reduce costs in testing and integration for HNG-X, by combining objectives, merging test stages, and working more collaboratively.

### 2.1.3 Requirements & Acceptance Criteria

In contrast with earlier intentions for HNG, where ‘Functional Equivalence’ had been a key goal (i.e. ensuring, function by function, that all the functionality of the existing Horizon Systems were reproduced in the re-engineered solution, using the existing systems as the benchmark), HNG-X aims to provide ‘Business Equivalence’ (i.e. the re-engineered solution will provide the equivalent business capabilities, using a statement of the Business Requirements and associated Acceptance Criteria as the benchmark).

A full set of Business Requirements are to be formally stated for HNG-X, covering both functional and non-functional aspects. These will employ Use Case Modelling techniques, and will be elaborated to System Use Case level, and developed using UML, Sequence Diagrams, Activity Diagrams, etc. as and where appropriate and held in a central repository under formal change control.

Associated Acceptance Criteria will be developed alongside these Requirements, to articulate precisely the criteria that must be met contractually. Where these are to be satisfied through testing, the Acceptance Criteria will be formulated as required test scenarios, to remove all ambiguity.

As the 'driving' requirements are more operational in nature, reflecting the revisions to the solution needed to deliver the TCO reductions necessary, then a full set of Operational Requirements (functional and non-functional) will also therefore be developed, in parallel with the Business Requirements.

### 2.1.4 Solution Architecture

This section provides a brief outline of the proposed solution architecture, from a testing and integration perspective. Please note – this is provided purely for the convenience of the reader, and is intended to be illustrative and not definitive. For a definitive view of the architecture the reader is advised to reference the definitive documentation for this area [6].

#### 2.1.4.1 Infrastructure

HNG-X will involve significant and widespread renewal/re-configuration across the whole spectrum of the supporting Infrastructure Systems. At the time of writing the precise changes are unknown, but the following list of likely areas of change should be considered for impact on testing and integration for HNG-X:

- Software Distribution
- Systems Management
    - Event Monitoring
    - Platform Management
    - Network Management
- File Transfer Management
- Asset Management
- Security
    - Access Control
    - Key Management

  - Secure Builds
- Network Systems
  - Branch LAN & Branch Router (& hub)
  - multiple Public Network WANs and associated Switches & Routers
  - Firewalls
  - Data Centre LANs and SANs
  - Inter Campus WAN and SRDF links
- Resilience and Recovery mechanisms (various levels)
  - Server
  - Application
  - Data
  - Transaction
- Disaster Recover mechanisms
- Peripheral Management
- Time Synchronisation
- End of Day Coordination
- Batch Scheduling

#### 2.1.4.2 Counter Layer

A new counter layer will be introduced, entirely replacing the existing Horizon Counter systems, though reusing their existing hardware platforms:

- The existing counter PC platforms will be retained, together with the attendant peripherals - Pin Pad & Card Reader, Bar Code Scanner, Scales, Slip/Tally Roll Printer, and Back Office Printer. (Note: The potential replacement of the Slip/Tally Roll Printer, which is being discussed at the time of writing this strategy, is assumed to be outside the scope of HNG-X and so is not considered here.)
- A new Branch Router will be installed at each branch, in advance of the main HNG-X rollout. Initially this will emulate the existing arrangements allowing the gateway counter to effect all communications with the centres as usual. When the branch is migrated to HNG-X during rollout, the new HNG-X counters will each communicate independently with the centre via the Branch Router.

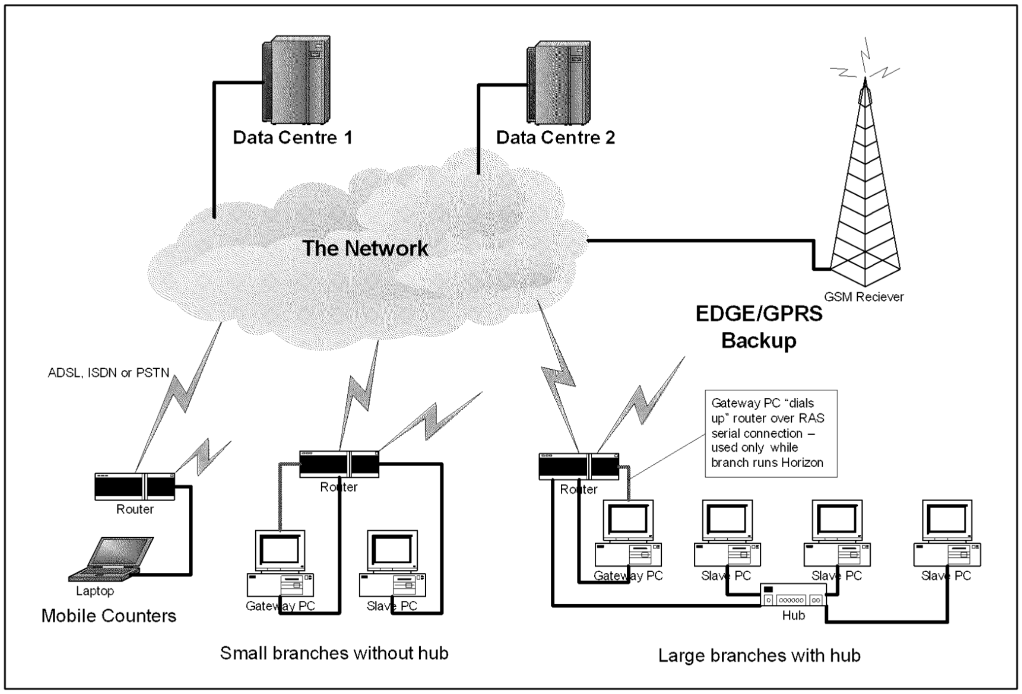


Figure 4 : Use of Branch Routers

- The existing Windows NT4 operating system will initially be retained, and then later, following successful rollout of the HNG-X counter applications, this will be migrated to Windows XP.
- The existing counter applications, the Riposte Desktop with its proprietary GUI/API structure and peripheral management system, and the Riposte Message Server with its self-replicating local data store and message transport system, are all to be entirely replaced.
- The HNG-X counter applications will adopt a Client Server architecture, with Fat Clients at the counter communicating with Application Services at the centre (in a Web Services framework). The Fat Clients will employ data driven, object based application components, orchestrated by a generic process engine, all developed in JAVA.

#### 2.1.4.3 Web Services Layer

A new Web Services layer will be introduced to provide the server-side processing for the counter application clients:

- A new Web Services layer will be developed, probably using the InterStage product as the framework to orchestrate the new Application Services developed in J2EE.
- All onward processing of counter client requests to Interactive Agents and to the Branch Database will be co-ordinated by the Web Services layer.
- New Session Management and Peripheral Management facilities will be developed accordingly.

#### 2.1.4.4 Branch Database

A new centralised transaction data store will be introduced, entirely replacing the existing Correspondence Server systems:

- The existing clusters of Correspondence Servers, with associated Riposte Message Server and self-replicating Riposte Message Stores, will all be entirely replaced.
- A Branch Database will be developed to provide all necessary storage of transaction history. (Note – with Client Server architecture, no longer a need for local transaction store at the counter either.) Relatively little code development envisaged, probably in PL/SQL.
- The Audit Server system will be amended to source the audit data from the Branch Database system.

  (The following schematic illustrates the various logical sub-divisions of the Branch Database.)

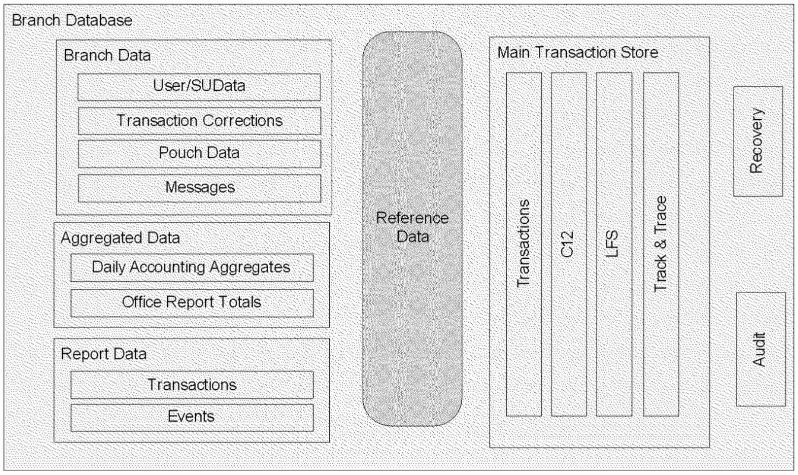


Figure 5 : Branch Database – Logical Sub-divisions

- An important aspect of the Branch Database is transaction throughput capacity. Being at the very heart of almost all HNG-X Data Centre processing, this is a performance critical area, and will be developed using ORACLE RAC technology to provide the necessary throughput capacity and potential scalability. In order to avoid excessive RAC interconnect traffic (where different server nodes share cached I/O), which is the most common degradation effect in RAC configurations, it is intended to physically partition the transaction data such that data for a given branch will always reside on a single node (not dissimilar to the way in which the existing transaction data is partitioned across 4 separate clusters of Correspondence Servers). This will in fact eliminate RAC interconnect traffic. The following schematic illustrates the principle.

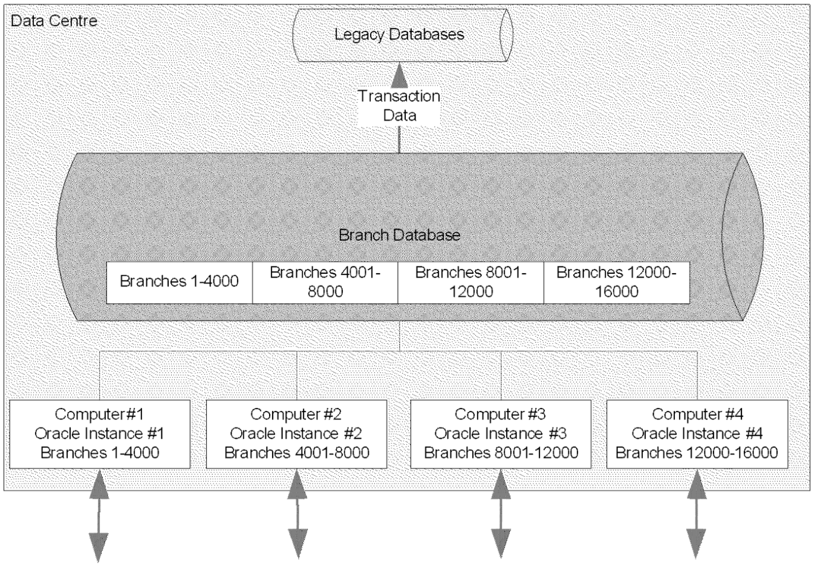


**Figure 6 : Branch Database – Logical Partitioning (Chunking)**

### 2.1.4.5 Agent Layer

The Agent systems will be amended to recognise the new Branch Database in place of the existing Correspondence Server systems:

- Bulk Loader Agents remain functionally identical, the only necessary change being to format and insert the incoming data stream into ORACLE Tables in the new Branch Database instead of writing corresponding messages into the Riposte Message Store on the Correspondence Servers.
- Bulk Harvester Agents are likely to become obsolete, with the Host's Database views being redirected to point at the Tables in the new Branch Database instead of the ones currently populated by the Bulk Harvesters. In

any event, the underlying data structures remain the same from the Host perspective.

- Interactive Agents are rather more complex, and whilst it seems likely that the majority of change necessary for HNG-X in this area will once again be concentrated on the new interface to the Branch Database instead of the existing Correspondence Servers, it also seems likely that some further changes will be required to adapt the detailed internal mechanisms accordingly. In particular, they may be impacted by the changes required to the supporting infrastructure (e.g. the current heart-beat mechanism may be impacted by the change in resilience model).

#### 2.1.4.6 Host Layer

No change to the Host system functionality is planned. The databases may, however, be re-platformed to share the Branch Database infrastructure (ORACLE RAC Servers and common disk arrays) in order to reduce ongoing management costs.

#### 2.1.4.7 External Interface Layer

No changes to the external interfaces are planned.

#### 2.1.4.8 Disaster Recovery Provision

HNG-X will move away from the complex configuration employed by Horizon, known as Active/Active, and instead adopt the more customary configuration where one primary site serves Production, and the other remains in reserve for use in Disaster Recovery (DR).

It is intended to make use of the DR equipment, during normal operation, for testing purposes, to obviate the need for a separate and expensive testing estate as is currently in place for Horizon testing. (This alternative usage is not possible for Horizon as both Data Centres are used for Production workloads in Horizon's Active/Active configuration.)

(See 2.1.5 below for more detail.)

### 2.1.5 Data Centre Strategy

The Horizon system operates two Data Centres, currently known as Wigan and Bootle, named after the towns where they are located. Two sites are needed in order to maintain business continuity in the event of a catastrophic failure at one of the sites. Horizon employs a complex configuration, known as Active/Active, where both Data Centres operate as Production sites (i.e. processing Live business transactions), but with each running at below half capacity (i.e. both sites are sized to each be able to accommodate full projected peak loads). In the event that either site suffers a catastrophic failure, then the other site is able to continue, re-routing the whole workload to it.

For HNG-X it is planned to move away from the complex Active/Active configuration employed for Horizon, and instead adopt the more customary Active/Standby configuration. This configuration still requires two sites in order to provide business continuity in the event of a catastrophic failure. However, the configuration is much simpler – only one of the sites (the primary site) operates as the Production site, whilst the other site remains in reserve as a Disaster Recovery (DR) site, only to be activated in the event of a catastrophic failure of the Production site. Again, each site is sized to accommodate full projected peak workloads.

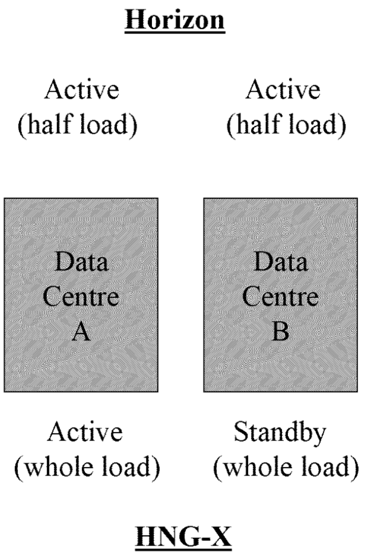


**Figure 7 : DR Configurations – Active/Active vs Active/Standby**

N.B. as no standard terminology exists for such configurations, this Active/Standby configuration may also be seen referred to elsewhere as being Production/Standby, Active/DR, or Production/DR. These should all be taken as being interchangeable.

This revised topography is desirable partly to simplify the solution, and partly to make more efficient use of the equipment – in common with many organisations HNG-X plans to exploit the Standby site, when not in use for DR, to provide equipment for testing. In the long term this would obviate the need for the very extensive testing estate required currently for Horizon. This represents a significant saving, not just in terms of raw hardware, but more significantly in terms of the time and effort required to manage and maintain this additional estate.

This departure from current operation involves a number of changes – hardware usage (platform configuration and re-configuration), data synchronisation and replacement (Production to DR, and separate storage for testing), mode switching

(Production routing to DR routing, and Testing use to DR use), the underlying Resilience and Recovery model, and the Operational and Business Procedures for DR. As with all changes, they demand validation.

### 2.1.6 Release Structure

N.B. At the time of writing the Migration Strategies are available only in outline (see [7] & [8]), but these provide a sufficient overview of the key elements of the Release Structure to inform the testing strategy at a high level. However, it must be remembered that the following only reflects the current thinking at the time of writing, and not a baseline position. It must be revisited once the Migration Strategy is formally issued. Also, the material below describing the migration is included in this document for the convenience of the reader, and should not be taken as definitive. For a definitive view the reder is directed to the relevant source documents [7] & [8].

#### 2.1.6.1 Outline Structure

The Release Structure can be illustrated in outline by the following chronology:

- Advance Preparation in Horizon (not in any particular order)
    - Move Horizon Data Centres to new locations, and install additional hardware to support HNG-X at both sites in advance of HNG-X
    - Install Branch Routers at Branches, again well in advance of HNG-X
    - Install additional equipment needed for HNG-X at both data centres, again well in advance of HNG-X migration (can be used for testing purposes)
- Migrate Data Centres to Hybrid Horizon/HNG-X state
- Rehearse revised DR arrangements
- Pilot HNG-X Go-Live, migrating and cutting over at first just one branch, then a handful, and in all 200-300, with the Pilot planned to last some 6 months
- Full HNG-X Rollout – continue to migrate and cut over the remainder of the branch network (about 14000), over a period of a further 6 months.
- Migrate counter operating system from Windows NT4 to Windows XP, over a period of a further 6 months.

**Figure 8 : Outline Release Structure**

#### 2.1.6.2 Data Centre Migration

The two existing Horizon Data Centres are planned to be relocated to two new sites well in advance of HNG-X deployment. The details are not currently known, and are largely irrelevant for the purposes of setting this testing strategy. We will simply refer to them as Data Centre 1 (DC1) and Data Centre 2 (DC2).

The new hardware necessary to support HNG-X will be installed at both sites alongside the Horizon hardware, and will all initially be available for testing purposes.

The HNG-X Data Centre systems will be deployed and configured in a hybrid fashion, with both Horizon and HNG-X systems operating alongside each other. This 'Dual-Running' configuration will continue until such time as the entire network of branches has been migrated and cut over to HNG-X.

Assuming that the amended Agent systems for HNG-X are created separate and distinct from the existing Horizon Agent systems, it is envisaged that the central systems up to and including the Agent layer will co-exist side by side, whilst the Host systems will form a common external facing layer for both Horizon and HNG-X, effectively fusing the two data streams.

(To clarify, if it is decided instead that when amending the Agents for HNG-X, a single common version is created, able to act for both Horizon and HNG-X, then this will become the point where the two data streams fuse, and both the Agents and the Hosts will form the external facing layer.)

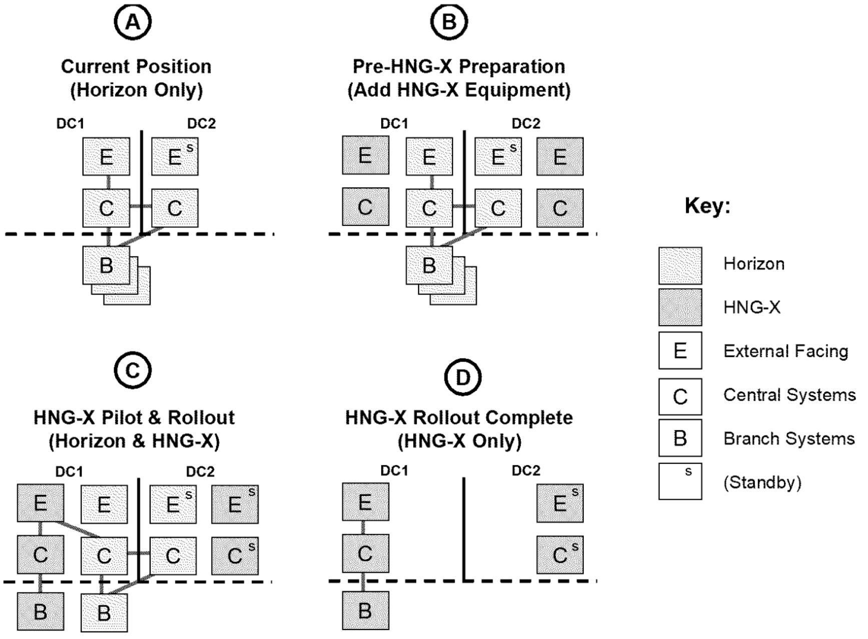


**Figure 9 : Data Centre Transition through Hybrid Configuration**

Transaction data (including history) is migrated to the Branch Database, branch by branch, in synchronisation with the migration of branch systems (see 2.1.6.3 below).

### 2.1.6.3 Branch (Counter) Migration

The Branch Routers will be installed in the branches in advance of the main HNG-X Counter deployment. Initially they will be configured to allow the existing Horizon Gateway PCs to continue controlling communication with the Data Centres in the current fashion. For larger branches with several counter positions, a hub will also be configured alongside the Branch Router to provide sufficient LAN connectivity.

After the Data Centres have been migrated and cut over to HNG-X ready for 'Dual-Running' (see 2.1.6.2 above) then HNG-X rollout can commence for the Pilot.

At first a single branch, then a handful, and finally 200-300 branches will be rolled out to form the Pilot, as follows:

- Stock position secured by End of Day processing, and replicated in the normal way to the Correspondence Servers.
- Position harvested to TPS (enhanced Agent and new database Tables).
- Overnight extract/load to Branch Database
- Overnight software migration to HNG-X counter systems
- Next day branch starts to operate under HNG-X

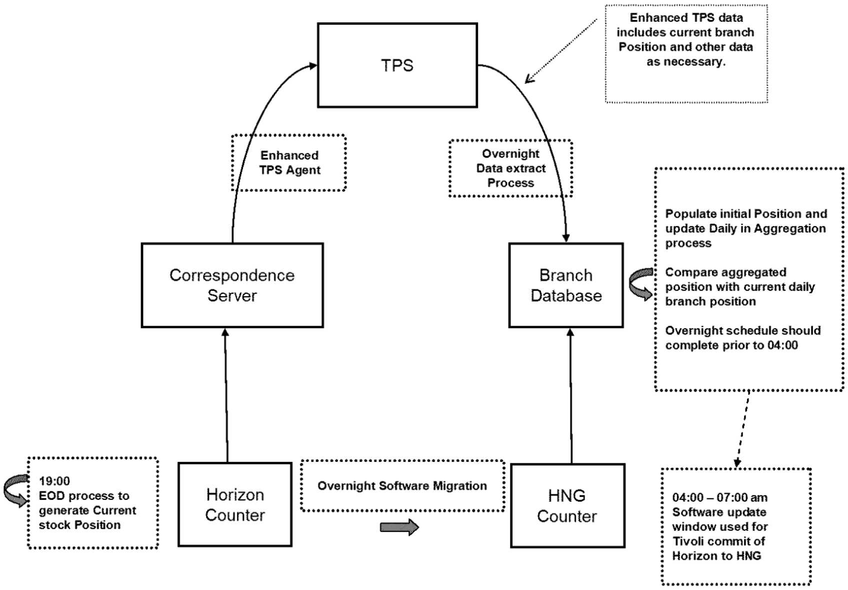

Enhanced TPS data includes current branch Position and other data as necessary.
TPS
Enhanced TPS Agent
Overnight Data extract Process
Populate initial Position and update Daily in Aggregation process
Compare aggregated position with current daily branch position
Overnight schedule should complete prior to 04:00
Correspondence Server
Branch Database
19:00 EOD process to generate Current stock Position
Horizon Counter
Overnight Software Migration
HNG Counter
04:00 – 07:00 am Software update window used for Tivoli commit of Horizon to HNG

**Figure 10 : Migration of Branch Data**

Following a successful Pilot (of about 6 months), the remainder of the branch network (about 14000) will gradually be rolled out in the same fashion, over the following 6 months.

The migration of the Counter Platform operating system from Windows NT4 to Windows XP takes place as a separate HNG-X release and rollout (again of about 6 months duration), following completion of the main rollout.

## 2.2 Implications for Testing & Integration

There are very many implications arising from the above considerations for testing and integration. Some are obvious, some less so. The most important, which serve to shape the necessary approach for testing HNG-X, are explained below.

### 2.2.1 No Reprieve … yet (Heavy Duty Testing)

The Horizon systems remain, as an integral part of the HNG-X solution, throughout the period Dual-Running (i.e. until rollout is complete across the whole branch network). It follows therefore that all the complexity, and the fragile inter-dependencies so familiar with Horizon, will still need to be dealt with:

- Localised impact cannot be assumed – changes are likely to extend and reach into other (often unexpected) parts of the solution – extensive, broad-based regression testing will continue to be required
- Heavy-duty, waterfall based (V-diagram) integration and validation remains appropriate
- Progressive (stage by stage) integration remains necessary (components, to sub-assemblies, to systems, to whole solution)

### 2.2.2 Iterative Development & Integrated Project Teams

Iterative development techniques will be adopted where appropriate (e.g in developing the GUI), and in any case an object-based methodology will be followed.

This demands close-working, integrated, mixed discipline project teams, with designers, developers, and testers working together throughout the development stage.

### 2.2.3 Collaborative Working & Merging Test Stages

Consistent with the drive to reduce TCO, it is important that duplicated effort is removed, synergy is exploited, and unnecessary separation of testing into multiple stages (causing increased costs and timescales) is avoided.

A common definition and prioritisation of testing objectives is required, and collaboration on achieving these objectives, between all parties concerned.

Test Cycles (for thread-based testing) must be planned and implemented in a flexible fashion, to better target priority areas without being rigidly tied to executing whole cycles on every occasion (i.e. the number and duration of test cycles should be driven by the prevailing circumstances, to best achieve the agreed objectives in the optimum fashion).

### 2.2.4 Robust Component Level Verification becomes Essential

The new HNG-X systems (Client/Counter and Web Services layers) are characterised by having a generic, data driven, object-based structure, with a highly flexible configuration. To best capitalise on this, both for HNG-X, and for the future, it is imperative that an extremely robust component base be developed, which can then be exploited to insulate the wider solution from the effects of localised changes, providing the basis for targeted testing around that localised change, and avoiding the need for so much widespread regression:

- Exhaustive, object-based, generic testing at component level
    - Test components in isolation
    - Prove object encapsulation
    - Use stubs and harnesses
    - Integrate modular hierarchies where they apply
    - Treat components on generic basis, and use wide spectrum of data to cover all allowed types/combinations (e.g. do not use just one instance of reference data!)
    - Where reuse, or common use is intended, prove both generically and by specific instance
    - Prove the direct integration of each component with its neighbours (including supporting infrastructure components) in the same fashion
    - Include appropriate non-functional aspects at component level (for example prove individual component performance budgets as specified in the detailed design specification)
- Comprehensive Regression Test packs created at component level, to effectively insulate the rest of the solution from later change (gradually building in the ability to localise change and target testing accordingly)
    - Requires formal planning and preparation of tests (Plans, Scripts, Data, etc.)

  - Preferably with automated test scripts to reduce the cost and time taken to execute regression test packs.

- It is assumed that the CM system will be extended to provide the necessary detailed and structural support for object-based development and testing, including associated impact analysis facilities to enable targeting of tests for localised changes. The testing requirements for this area (in addition to those of development and other parts of the programme) must be specified in parallel with the main business and operational requirements to allow sufficient time for these extensions to be made.

### 2.2.5 Non-Functional Aspects – Can't be Taken for Granted

With Horizon having been operating robustly for many years, with the security policy being put to bed long ago, and with much of the infrastructure having been in place and operating smoothly for release after release, it is natural to take many of the non-functional aspects of Horizon for granted.

However, for HNG-X, with everything except the Host and Agent layers being entirely re-engineered, and with widespread revision of the supporting infrastructure across the whole solution, the full spectrum of non-functional tests will once again need to be revisited for HNG-X.

This is an important factor to be considered, particularly in areas of the programme that may be resourced from the pool of existing Horizon personnel, where there could otherwise be a pre-disposition to take the non-functional aspects for granted

#### 2.2.5.1 Performance

The new Counter system, the Web Services Layer, and the Branch Database all figure prominently in the overall performance of the new HNG-X solution. The Counter system (or more specifically, an agreed selection of transaction types on the counter system) will need to be re-benchmarked to ensure acceptable performance at the point of sale. The Web Services layer will require careful configuration and tuning to avoid bottlenecks and adverse housekeeping (e.g. efficient garbage collection), and the new Branch Database is performance critical, needing to achieve the requisite transaction throughput, and to be demonstrably scaleable. Early indications of unit resource consumption, likely capacity, and behaviour under stress will all be required. Likewise, it is important that the scalability mechanisms planned for the new Branch Database be proven during the design stage, whilst corrections may be made to rectify any major issues. Full-scale performance, volume and stress testing will be required prior to embarking on full rollout. The gradual ramp-up of workload offered by the Pilot and then the Rollout period will help mitigate some performance risk, allowing the actual performance of the production system to be monitored, tracking performance with increasing workload and giving early warning of any major issues. The retention of the Host and External Interfaces intact (unchanged) also mitigates external performance impacts. Providing the internal performance of the solution is proven, and the volume/throughput rate across the external interfaces remains as is, then there will be no need to repeat full end-to-end performance tests involving the external systems.

#### 2.2.5.2 Integrity (Resilience & Recovery)

The full range of integrity testing will need to be performed for HNG-X, to verify and validate the new/revised resilience model, as it applies to each platform and system. The test set will be defined by taking inputs from the analysis of the potential points of failure, test objectives (based on risk via prioritisation) and identifying system, configuration and infrastructure changes.

The approach requires that integrity testing aspects are included from the earliest opportunity (at component level), and continue throughout the testing lifecycle. For example including detailed incremental testing of component failover, combined failover (combinations of hardware), network connectivity of failed over hardware, data recovery/synchronisation and end to end testing of failover items, leading to full data centre DR tests.

As many of the application systems are new, it is important that the system itself be used to confirm continuing integrity. Beyond simple component level testing, given the scale of change, it will not be realistic to combine much of this testing with mainstream functional tests. For HNG-X it is therefore intended to operate a separate test stream – Integrity Test - with associated resources and test envivronment(s) to focus on these aspects.

#### 2.2.5.3 Disaster Recovery

The Active/Standby model introduced with HNG-X means greater reliance on complex full data centre failover tests. The precise Disaster Recovery mechanisms must be fully proven, and each scenario rehearsed together with the supporting processes and procedures, prior to Pilot. These rehearsals should not be conducted by the testing teams, but rather will require the involvement of support personnel and other business-as-usual teams, emulating the circumstances of an actual DR event.

#### 2.2.5.4 Operability and Systems Management

Given the main driver, reduction of TCO, it will be important for HNG-X to include a full range of Operability tests, including validation of all the principal Systems Management facilities, to demonstrate that the desired savings in operating the solution can be realised as expected. It is normally practical to combine these tests with either the Integrity Testing stream or the later Functional Test streams in RV (or a combination of the two).

#### 2.2.5.5 Security

The approach requires that security testing aspects are included from the earliest opportunity (at component level), and continue throughout the testing lifecycle.

Security will be a defined property of all system components as specified by the Requirements and the agreed system security policies. At the individual component level the security requirements are unlikely be a major factor in the normal testing process since it tends to manifest itself until higher levels of integration.

From CIT onward, security testing becomes more significant: secure platform builds, database access controls, network security, data centre firewalls, antivirus software (both data centre and counter platforms), intruder detection etc. Security testing aspects will be included accordingly in each stage, and so also be covered by the corresponding Regression Test packs.

At the application level session security for counter applications by passwords will be new to HNG-x (VPN from branch to data centre not being retained from Horizon) and hence will require extensive testing. Other non-functional aspects of security such as the system management functions (e.g. secure access to platforms for support purposes) will be tested during System Test and in the SV&I stage.

Where particular tests, perhaps by virtue of the sensitivities surrounding secure materials (e.g. Live keys, encryption algorithms, etc.), require specialised or highly secure test environments, then it will not be practicable to combine such tests with the mainstream threads, and these will be dealt with in a separate stream of testing – Security Test. However, care should be taken avoid bundling all security aspects into this area.

(TBC when the security requirements have been agreed and the revised access security policy and security specification become available.)

#### 2.2.5.6 Usability

The entire counter application layer is being replaced for HNG-X, moving away from the proprietary GUI provided by the Riposte Desktop systems, and with a conscious intention to revise the user interface to improve efficiency where the opportunity presents itself. Certainly therefore the user interface will require full verification and validation.

There are three primary aspects to this:

- Ensuring that an appropriate UI definition is specified (UI Style Guide Development).
- Confirming that the agreed definition is correctly implemented (UI Functional Conformance)
- Identifying potential usability issues which may remain in the completed solution prior to Live use, so that they might be mitigated and managed to

avoid unnecessary business impact, and to determine whether future UI changes may be required (Usability Testing)

A detailed approach for this area is being produced separately and the description included here is for the convenience of the reader only and should not be taken as definitive. For a definitive view the reader is directed to [12]. This area is planned to proceed in 4 stages, each with their own associated assurance and acceptance activities:

- Stage 1 – Identification of Logical UI Requirements. (TBC when [12] has been completed and agreed).
- Stage 2 – Definition of Logical UI Constructs (TBC when [12] has been completed and agreed).
- Stage 3 - Definition of Rendered UI Constructs (TBC when [12] has been completed and agreed).
- Stage 4 – Realisation and Assurance (TBC when [12] has been completed and agreed).

(TBC when the UI requirements are completed and the associated assurance and acceptance processes for this area have been determined and agreed.)

### 2.2.6 Migration Complex and Critical

The system and data migrations required for HNG-X, both at the Data Centres, and at the branches (which continues branch by branch throughout the rollout period), are absolutely fundamental to the success of the deployment. It is a complex area requiring careful and detailed planning. Thorough verification and validation will be essential.

Individual migration-specific components must be verified with the same thoroughness and attention to detail (at the component level) as for the mainstream application and infrastructure products being developed for HNG-X, in order to trap defects as early as possible.

Migrated data should be introduced into the mainstream tests as soon as practicable, interleaving migration tests with functional test cycles.

Full blown rehearsals of the detailed migration plans must be completed prior to Pilot.

### 2.2.7 New/Revised Processes & Tools

To minimise the impact and maximise the benefit of iterative and targeted testing, using comprehensive regression testing packs, it is a requirement of this approach to adopt automated testing techniques wherever and whenever it provides a clear benefit. The adoption must therefore be pragmatic, embracing automation where it can be readily applied, and avoiding it where costs become prohibitive. The detailed test automation strategy will need to be developed, but it is clear that the primary focus should be in the early verification stages (these are small, self contained, and have fewer data dependencies to contend with, but require exhaustive coverage, making them ideal candidates for automation). Automation of the later stages of testing should be approach more critically, adopting automated scripts only where the benefit is clear, and where the cost of future maintenance of scripts will not escalate. (Sample coverage to support minimum regression testing would seem appropriate.)

The new testing processes and tools necessary to address the Object-based developments and their new technologies (including any test automation involved) must be developed and piloted well in advance of their full-scale use in anger.

Similarly, the mechanisms necessary for exploiting the DR for testing purposes, and so to obviate the need for a separate Test Estate, must be devised, developed, piloted and deployed as early as practicable, and in any event prior to Pilot.

## 2.3 Approach Required

Based on the above considerations and implications, an appropriate testing approach for HNG-X is in outline as follows:

- Engage with the Requirements Analysis and Systems Analysis team(s) at the earliest opportunity to influence the emerging requirements, ensure that they are testable and to assist with the codification of appropriate scenario based Acceptance Criteria. (It is important to note here that requirements traceability will have to be maintained throughout, from requirements sign-off through to acceptance sign-off, and so the means by which this traceability will be achieved must be built into the requirements specification process from the outset. Some method of persistent and unique identification will be essential.)
- Run an Architecture Proof of Concept (APOC) exercise to provide early feedback on the suitability of the proposed architecture, and to prove the scalability mechanisms and get early indications of the likely system performance. (Commence Performance Modelling to support this activity.)
- Continue Performance Modelling throughout HNG-X.
- Run a Testing Proof of Concept (TPOC) exercise to develop and prove the necessary testing processes and tools to cope with the object-based development and the new technologies used for HNG-X, and to pilot the mechanisms for making use of the DR site as the Test Estate. (As the TPOC will be run alongside the APOC, then it is likely that certain objectives for the TPOC may in fact be achieved by the APOC. Care should be taken in planning the two to exploit synergy and avoid duplicated effort.)

- Adopt Risk Based Testing, to ensure that the principal drivers of reduced TCO and Business Equivalence are achieved in an acceptable and cost effective fashion.
- Apply Objective Driven Testing techniques to simplify the implementation of Risk Based Testing, and to facilitate removing duplicated effort, exploiting synergy, reducing overall elapsed time, and maintaining product quality.
    - Analyse source material, derive Test Objectives (common set covering needs of all parties)
    - Engage with Business and Operations to conduct highly granular risk assessments (workshops), and so prioritise the Test Objectives
    - Start the outline test planning by seeking to exploit synergy, combining multiple objectives to be covered by a minimum number of tests
    - Normalise the outline tests by considering their likely environmental needs/constraints
    - Produce High Level Test Plans accordingly (Logical level)
    - Generate Low Level Test Scripts (ideally in automated form) and associated Test Data.

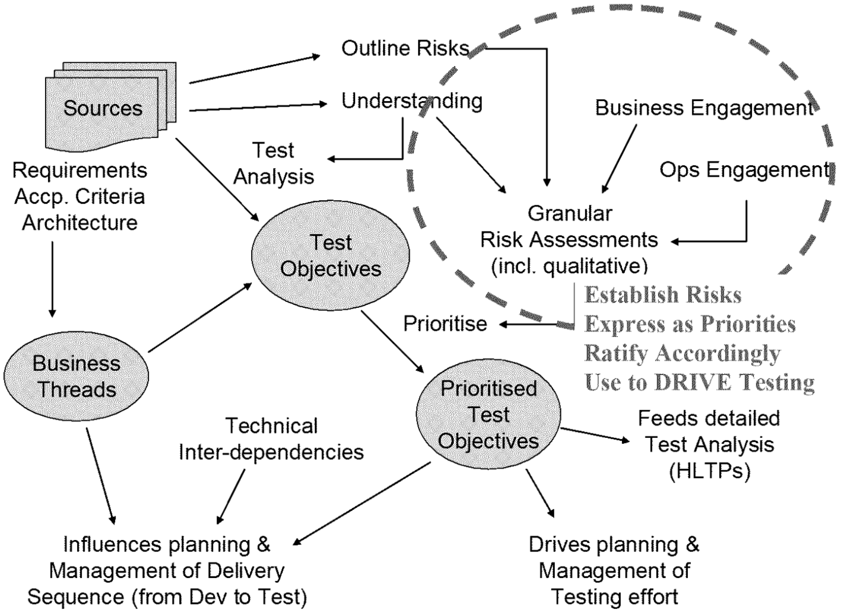


**Figure 11 : Objective Driven Testing - Outline**

- Apply a process of Progressive Integration, incrementally assembling and proving ever larger portions of the solution, following a clear sequence of lifecycle stages (i.e. adopting appropriate controls regarding the promotion of deliverables from stage to stage to ensure their inherent inter-dependencies will be satisfied).

- For existing Horizon areas being carried forward into the HNG-X solution, follow the existing testing lifecycle, but create and retain comprehensive Regression Test packs for the Unit Test (UT) and Extended Link Test (ELT) stages.
- For new developments, take a generic perspective to provide exhaustive verification (against the design spec) at the component level.
- Increase the emphasis on Code Reviews to trap code defects at the earliest possible opportunity. They also serve to apply an additional perspective (testing) on the design specification.
- Adopt new terminology to emphasise the need for increased rigour in this area – Component Test (CT) and Component Integration Test (CIT).
- Assemble and validate (against the requirements) each system as and when it becomes viable, and integrate with its supporting infrastructure – System Test (ST)
- Adopt integrated, mixed-discipline project teams for CT, CIT, & ST.
- Make extensive use of stubs and harnesses, etc., for CT & CIT, and to a lesser extent for ST.
- Verify all system interfaces against their design
- Formally plan and prepare all tests (Plans, Scripts, Data) and preserve results, to provide a comprehensive set of Regression Test packs, ideally automated.
- Promote flexible planning and pragmatic implementation to focus on high priority areas first.
    - Plan test cycles (thread based tests) flexibly, with high priority objectives to the fore.
    - Run test cycles pragmatically, being prepared to terminate the cycle early to progress fixes on high priority material first, and rerun in quick succession as appropriate.
    - Must have quick and easy mechanism for resetting environments (hours not days).
- Utilise Entry & Exit Criteria (and Suspension & Resumption Criteria) to provide realistic Quality Gates for controlling testing progress (from stage to stage) on the basis of objectives achieved rather than cycles run or arbitrary planning dates/deadlines reached.
- Pass proven systems to Independent Test area as and when each has become sufficiently stable (Entry Criteria) for onward integration and validation – System Validation & Integration (SV&I).
- Formally validate each external interface (DIT) as a collaborative activity
- Iteratively validate the emerging solution and integrate with the supporting infrastructure, until the entire solution has been assembled and fully validated, running End to End data flows within the bounds of the HNG-X solution. (Thread based testing, multiple test cycles, progressively introduce migrated

data.) Threads should pre-empt later reuse in RV for wider End to End running.

- Separate from SV&I conduct full Performance, Volume, and Stress testing
- Separate from SV&I conduct full Integrity testing.
- Extend End to End data flows out into external systems, and interleave with full blown rehearsals of migration plans – Release Validation (RV)
- Treat the final cycle of RV as the formal UAT
- Adopt a collaborative working approach throughout
    - Establish a core team composed roughly 50% Fujitsu Services Testing and 50% Post Office Ltd Testing personnel.
    - Use core team as the primary resource for engaging with the Requirements and Systems Analysis team(s), and for conducting the early test planning (see Objective Driven Testing bullet above).
    - Go on to use the core team throughout the project to provide the necessary strategic and technical direction and assurance.
    - Introduce additional Post Office Ltd Testing personnel and Operational personnel into SV&I, integrated into the Fujitsu Services managed test teams, to both contribute business/operational knowledge, to provide extra resource, and to gain knowledge of the new systems.
    - Introduce further Post Office Ltd Testing personnel, including end users, into RV.
- Pass through Release Authorisation Board (RAB) into Pilot (hand system over to Operations, Pilot run by Post Office Ltd.

The following schematic summarises the outline approach, illustrating the lifecycle stages involved and their main areas of coverage.

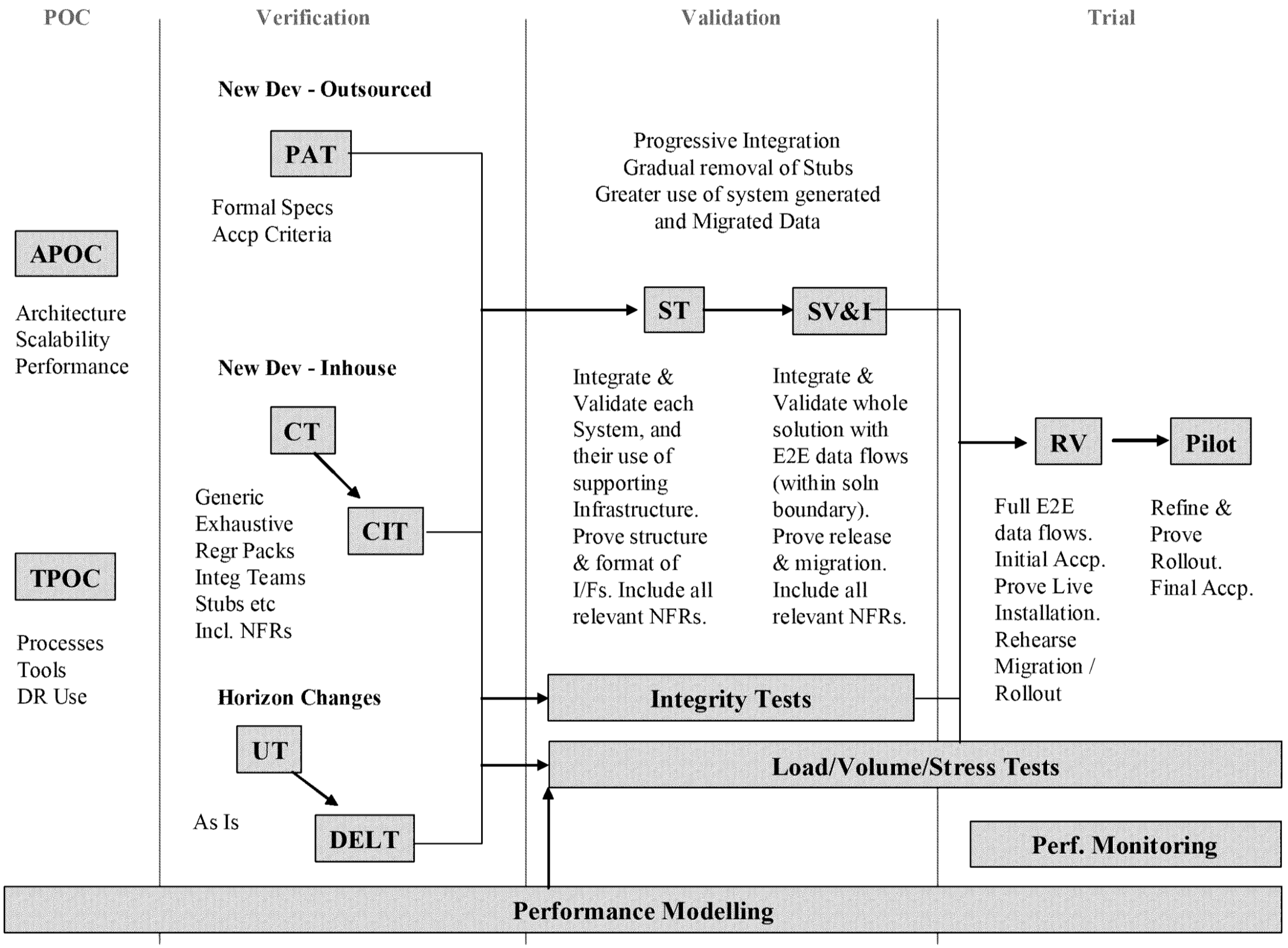


Figure 12 : Outline Approach

# 3.0 Goals & Objectives

## 3.1 Test Mission (Top level testing goals)

The following defines the mission for the overall testing and integration effort for the whole HNG-X programme.

- Provide exhaustive yet realistic verification and validation of the entire solution prior to Live release
- Govern the testing effort on the basis of risk, using the resulting prioritisation of the detailed testing objectives to drive the process.
- Trap defects as early as possible in the lifecycle, placing a heavy emphasis on Reviews, CT, and CIT, with comprehensive (and preferably automated) Regression Test packs.
- Demonstrate that the principal drivers of the programme have been met, by running tests satisfying all those Acceptance Criteria specified as to be met by testing.
- Provide management information (feedback on testing progress and outcome) to the business.
- Establish a comprehensive Performance Management regime, and in particular conduct extensive Performance Modelling from an early stage and on an ongoing basis.
- Develop efficient and effective testing processes, aligned with the new generic object-based developments, the desire to adopt a more collaborative style of working, and the move toward using the DR site as a Test Estate. Prove these, and their supporting tools and automation, through a series TPOCs well in advance of their widespread use for mainstream testing.
- Exploit stubs, harnesses, emulators, and the like, to provide a sound basis for component level testing in isolation, and to enable limited integration as early as possible in the lifecycle.

## 3.2 Test Motivators (Key sources determining tests)

The following defines the primary motivators influencing the overall testing and integration effort for the whole HNG-X programme. This should be used as a checklist of necessary inputs for the high level test analysis, early planning, and later more detailed test analysis, planning and scripting activities.

(The following is a standard ‘vanilla’ list of principal test motivators for large programmes. This will need to be revised in the light of emerging information (in particular the Development Strategy, and the corresponding deliverables). Some of these will remain as stated, some will use alternate terms, and some will not be relevant for HNG-X and can be removed.)

- Vision
- System Use Cases (SUCs) and related Models (SUCMs)
- Supplementary (non-functional) Requirements (NFRs)
- Supplementary Usability Requirement
- Acceptance Criteria
- Business Processes, Procedures, Instructions
- Training Strategy
- Release Strategy
- Implementation/Migration Plans
- Software Architecture Document (SAD)
- Technical Environment Description (TED)
- Physical Environment Description (PED)
- Deployment Configuration Guide (DCG)
- Infrastructure High Level Design (IHLD)
- Analysis User Experience Model
- Logical User Interface Model
- Analysis Entity Class Model
- Logical Data model
- Use Case Realisation Sequence Diagrams
- Activity Diagrams
- State Diagrams
- Class Diagrams
- SRS (composite artefact gathering elements to forma comprehensive System Requirement Specification)
- Requirement Catalogue
- Business Rules Catalogue
- Product Catalogue
- Business Specified Test Points (BSTPs)
- Business Threads (BTs)

- Change Requests (CRs)
- Risk Register
- Programme Plan
- Delivery Plans
- Environmental Constraints
- Stakeholder Requests
- Test Reports (3rd Party Systems)
- Regression Test Packs (existing or 3rd Party systems)
- Known Error Logs
- Interface Specifications (EPIDDs)

## 3.3 High Level Objectives (The Principles)

The following list of high level objectives represent the fundamental principles to be observed throughout HNG-X testing and integration:

- Test early – incremental testing (test what you can at any given stage/time)
- Developers are responsible for writing tests for the code they create
- Test enough (risk/cost based, value for money)
- Independent Testing Unit to integrate and validate whole solution
- Progressive integration – component level, then system level, then solution level, then release level
- Ability to provide full requirement traceability through tests being linked to acceptance criteria via Business Threads
- Optimise the use of automation to reduce cost and improve quality – focus primarily on component level verification for maximum payback
- Insulate test investment from change impact – exhaustive testing, with comprehensive regression test packs, for component level verification
- Targeted testing (avoid blanket regression)
- Test environments appropriate for tests concerned – highly synthetic for component level verification, through to being as close to Live 'shape' as practical for validation and integration the whole solution, and being of significant scale for performance, volume, and stress testing – all live components to be trialled on target Live platforms before released to Live

- Consistent use of test management tools across all testing streams – exploit synergy – reuse test materials wherever practicable
- Testing teams working collaboratively with both Post Office Ltd Testing and Fujitsu Services Operations – combine test objectives, avoid duplication of effort, and exploit synergy – share and reuse test materials wherever practicable
- Test management tools to be fully integrated with other software development lifecycle tools (e.g. defect management, requirements management and change management systems).
- Testing fully integrated with software development lifecycle – early engagement, from Requirements Analysis onward – component level verification and system level validation conducted within integrated mixed-discipline project teams
- Early engagement with Post Office Ltd and Fujitsu Services Operations – risk assessments, prioritisation of test objectives
- Objective driven testing – DRIVE all testing activity on the basis of the prioritised test objectives, so observing the underlying risk assessments
- Progress through achievement – promote from stage to stage on the basis of the test objectives achieved, and so the ‘readiness’ of the system(s) under test to advance to the next stage of progressive integration. (i.e. risk-based, not time driven)
- Flexible test planning – bringing higher priority material to the fore, and organising testing cycles such that they can be run very flexibly, with rapid resetting of test environments to enable testing cycles to be restarted quickly and economically
- Continuous reassessment/reinforcement of strategic direction – joint core team working throughout to set and check direction and gain assurance
- Non-functional aspects included in every stage of testing

# 4.0 Strategic Approach

This section expands on the outline approach described in section 2.3 above.

# 4.1 Objective Driven Testing

Central to this strategic approach is to drive all aspects of the work (planning, setting the priorities, engineering the necessary tests, preparing the test materials, allocating the necessary resources, executing the tests and checking the results) based primarily on achieving the necessary objectives in an efficient and effective fashion.

This means that all the tests have to be justified by the requirements of the release. One of the primary objectives of the test analysis phase is to identify the required set of tests. The importance of each test is also assessed so the most critical ones are given priority.

## 4.1.1 Risk Based Foundation

Risk based testing is the desired foundation of many testing approaches. However, it is frequently implemented in too academic a fashion, without due regard for the real-world implications involved in testing a large and complex system. This approach takes care to recognise the needs of testing, whilst protecting the core objectives of a risk-based approach.

In the real world, where the aim is to achieve an acceptable result rather than a perfect one, exhaustive testing does not mean directly proving all possible permutations of all component elements in every permissible fashion. Budget and schedule (the costs) are important factors in the requirement. When considering what testing coverage must be achieved, and when each element of that coverage must be addressed in the lifecycle, the relative Cost must be assessed against the prevailing Risk. This Cost-v-Risk assessment is pivotal in planning and managing the assurance and testing effort.

Business input is vital. Post Office Ltd involvement in the initial risk assessment process, coupled with joint forums (such as PRFs) to continue providing business input on an ongoing basis, will ensure that a balanced perspective is obtained.

Risk identification, assessment, and management can be a complex and tortuous affair. At best testers find the discipline a difficult one.

A method developed to reduce some of the pain involved is Objective Driven Testing. This helps to streamline the assessment process and reduce the ongoing administrative burden, and so improves decision support.

### 4.1.2 Objective Driven Method

Objective Driven Testing is a method of performing risk assessment to govern testing effort, routinely and repeatedly, without having to directly assess the costs and risks on each occasion. This is achieved by recording a highly granular set of priorities based on an initial assessment of the costs and risks, and then using these relative priorities to govern testing effort, exploiting their implicit reflection of the underlying costs and risks.

The key is to construct a comprehensive matrix of relationships, in terms of detailed test objectives (in some methodologies referred to as ‘test points’), and using the inbuilt traceability of the test analysis and test preparation processes to relate these test objectives to the relevant requirement, analysis, and design artefacts.

An initial risk identification and risk assessment activity is performed, and the resulting scores are related to the test objectives via the traceability relationships, to grade their relative priorities. As the Test objectives form an integral part of the test analysis and test preparation process, then again the inbuilt traceability associates them with the test plans, reviews, test scripts, and test steps accordingly. So each potential unit of assurance and testing can be implicitly cost-v-risk assessed by simply using the priority values assigned.

There are a number of additional benefits accruing from this method:

- Test Managers and Testers can relate to these priorities intuitively as they are directly associated with the test materials they prepare and follow.
- They are more amenable to qualitative or subjective risk assessments as the relative priorities can just as easily be set subjectively
- The inbuilt traceability vastly simplifies assessing the impact of change, as any impact analysis process will already be following those same traceability relationships.
- Since reporting of test process is driven largely by the tests being run, and these can be directly related to the test objectives they satisfy, then it is a simple matter to report (albeit implicitly, in terms of priorities met) on residual risk exposure as a factor of test coverage.
- When risks change, then simply reassigning the priorities will automatically flow through to what tests should / should not be required.

### 4.1.3 The Principles

The main principles involved are:

- Analyse the available source documentation that defines the system(s) under test and determine WHAT needs to be tested (and WHY), cataloguing this in the form of Test Objectives, traceable back to the relevant requirements. (It is often helpful to conduct the analysis from a number of discrete perspectives –

Functional, Performance, Security, Integrity, etc. Whilst this may assist the process, for example improving target audiences for quality reviews, it is important that these perspectives are not permitted to influence the forward development of the tests.) Objectives must be:

- ✓ Meaningful – stated clearly and unambiguously
- ✓ Measurable – obvious what must be accomplished to satisfy
- ✓ Manageable – realistic, achievable, and not too complex

- Conduct fine grain risk assessments, engaging Fujitsu Services Operations and Post Office Ltd as appropriate, and so prioritise the Test Objectives accordingly, effectively indicating WHICH Test Objectives should take precedence over others. (The relative priorities assigned then become an implicit reflection of their associated risks.)
- Group these prioritised objectives in an optimum fashion, exploiting synergy, to accomplish multiple objectives at once, forming embryonic test plans (effectively indicating WHEN each objective will be addressed).
- Normalise these embryonic tests based on their environmental needs, and further combine them where appropriate to increase efficiency (effectively indicating WHERE the test will be performed).
- Engineer these as explicit tests to achieve the objectives, documenting HOW, (and finalising WHEN and WHERE) they will be performed, first planning the tests in detail at the logical level (HLTP) and defining the necessary supporting data, and then scripting them at the detailed physical level to provide clear instructions for their execution (LLTS or equivalent automated scripts).
- Throughout, maintain the traceability back through the objectives to the source documentation so that the rationale for each test remains clear. Ideally this is accomplished by using a CAST (Computer Aided Software Testing) tool with an integrated repository (such as Mercury TestDirector) to document the Test Objectives, HLTP and LLTSs, with appropriate interlinking relationships.

It should be noted that these principles apply iteratively, as further source documentation becomes available. Typically, business requirement-based sources are available earlier and are appropriate for early planning of solution level and release level testing, such as that performed within the SV&I and RV stages. As the system requirements emerge these can be further refined, and work can commence on planning the system level testing, such as that performed within the ST stage, whilst design-based sources become available a little later, and can be used to plan the component level testing and to refine the system level testing.

These principles should be embraced throughout the whole approach. To emphasise the point it is worth highlighting two common pitfalls to be avoided.

- Organisation – the management and staff organisation, and the various teams that may exist from time to time to perform different facets of integration and testing work, and the processes that they follow, should be considered as the means by which the required objectives may be achieved, not as a defining element of those objectives, and certainly not as a constraint. It is very easy for any organisation to become institutionalised, such that each constituent team simply 'does what that team does' irrespective of the extent to which it 'needs' to be done. When this happens, they lose sight of the objectives that must be achieved, and why. Their relative priorities are not considered. The process takes over, and each team is encouraged to operate more and more autonomously, separately analysing the work required of them. They become divergent, often duplicating effort, and, more seriously, gaps open up between them. The resulting test coverage achieved is inevitably going to be inappropriate, and the effort expended in one area compared to another becomes driven by the relative size of the 'teams' involved.

- Environments – test environments represent a significant investment, and quite rightly they have to be managed and protected. In some organisations they may be centrally managed, and in others they may 'belong' to particular testing teams. There is nothing wrong in this. However, it is critical that neither the limitations of any given test environment, nor their 'ownership' by a particular testing team, be allowed to influence what the testing objectives should be. Rather, they will influence How, When, and Where (and so possibly by Who), and maybe even If a given objective should be achieved. Test environments should not be regarded as static, 'cast in stone'. They should rather be evolved as necessary, designing them with the most cost-effective configurations, to best service the demands of the required test objectives. (It has to be said that test environments may constrain what can be achieved – some objectives may be proposed, which on further analysis may demand prohibitively expensive environments to fully achieve them. Their risk-based priority ratings will assist in deciding whether such objectives warrant the expense.)

One natural by-product of adopting a process of Objective Driven Testing is that strategic direction improves. By its very nature, it becomes clear (almost self-evident) to all involved exactly why particular tests should be engineered, and what facets of the system requirements/designs they are related to. Also, coverage analysis is made straightforward, and so duplication of effort, and gaps in test coverage, can be identified and avoided more easily.

### 4.1.4 Outline Process

The following schematic provides an overview of the way in which Objective Driven Testing is implemented within the POA approach to integration and testing. It illustrates the outline processes (Analyse, Prioritise, Optimise, Normalise, Engineer), giving an outline breakdown of each, highlighting the main activities and data flows. It also shows the relationship at each stage with the stated principles, which may be characterised by the simple questions:

- WHAT to test? – the Test Objectives
- WHY test them? – the rationale, usually relating back to Requirements
- WHICH take priority? – some objectives will be more urgent/important
- WHEN to test them? – grouping them in an optimal fashion
- WHERE to test them? – assigning an appropriate environment type
- HOW to test them? – engineering the tests accordingly

FUJ00232570

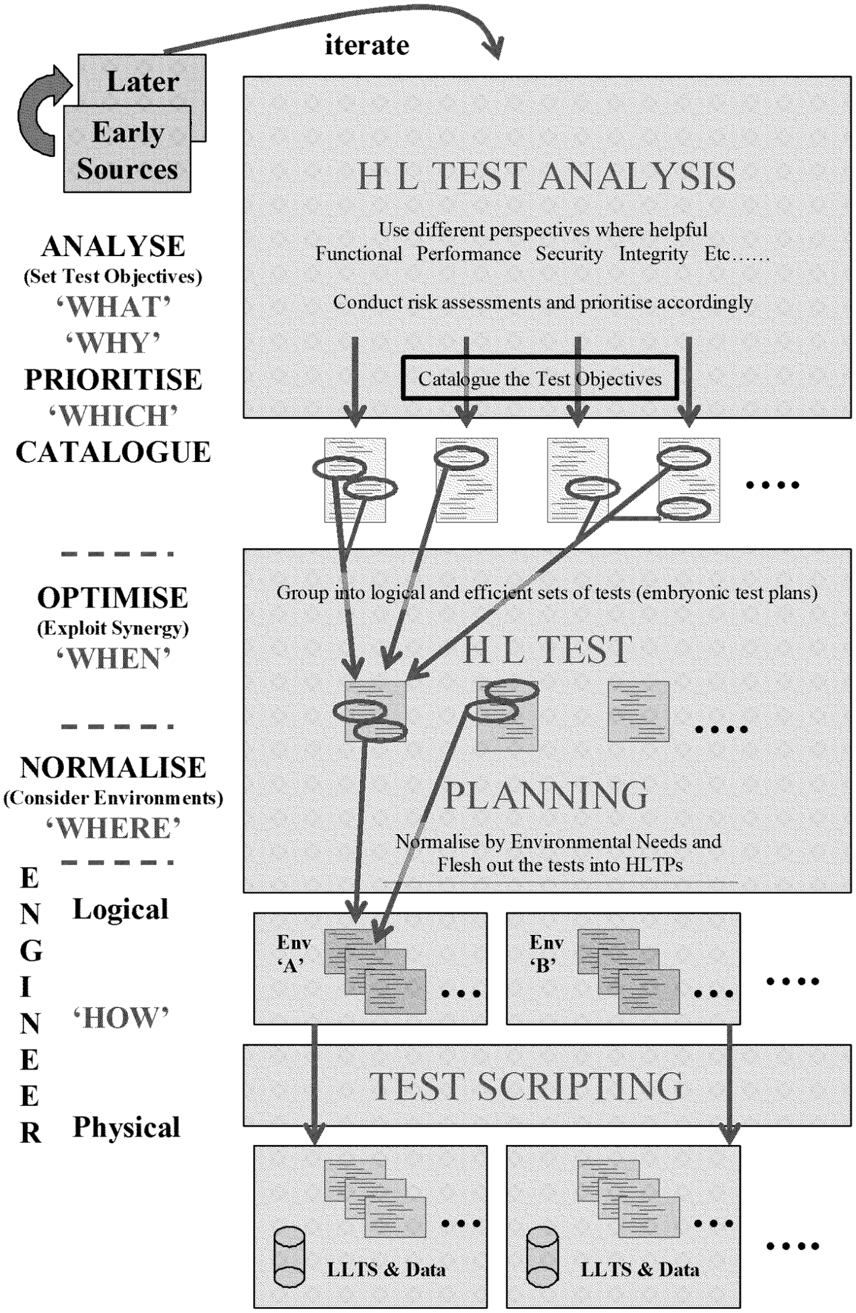

iterate
Later
Early
Sources
ANALYSE
(Set Test Objectives)
'WHAT'
'WHY'
PRIORITISE
'WHICH'
CATALOGUE
OPTIMISE
(Exploit Synergy)
'WHEN'
NORMALISE
(Consider Environments)
'WHERE'
E N G I N E E R
Logical
'HOW'
Physical
H L TEST ANALYSIS
Use different perspectives where helpful
Functional Performance Security Integrity Etc……
Conduct risk assessments and prioritise accordingly
Catalogue the Test Objectives
….
Group into logical and efficient sets of tests (embryonic test plans)
H L TEST
PLANNING
….
Normalise by Environmental Needs and
Flesh out the tests into HLTPs
Env 'A'
Env 'B'
…
….
TEST SCRIPTING
LLTS & Data
LLTS & Data
…
….

Figure 13 - Objective Driven Testing – Process Flow

## 4.2 Formal Planning & Preparation

Formally documenting test plans and scripts is an obvious consequence of adopting Objective Driven Testing (see 4.1 above), but is also essential for a number of other reasons, as follows:

- To provide traceability from the planned tests (and so the executed tests) back to the Requirements, as required by CMMI, and as necessary in order to conduct coverage analysis, and assess residual risk
- To make the tests reusable (re-runable) and so protect the considerable investment involved in engineering a correct set of tests for a large and complex solution such as HNG-X
- To enable impact analysis (and so targeted testing) based on changes to certain identified components, using the traceability provided by the formal plans and scripts
- To provide an ongoing, and easily understood and amendable testing infrastructure for future use
- To allow the creation (collectively) of comprehensive Regression Test packs

As implied in 4.1 above, the steps involved are:

- The prioritised test objectives (derived from the principal sources) are grouped into logical and efficient sets, optimising them to exploit synergy, and so producing embryonic test plans.
- Their particular environmental requirements/constraints are then considered, normalising them accordingly (effectively allocating them to the most appropriate and cost effective testing stage) ….
- …. and fleshed out further to form High Level Test Plans (HLTPs)
- These are at a logical level and so should remain independent of their implementation (e.g. any tools/automation used) and so form a valuable and enduring test asset
- Low Level Test Scripts (LLTSs) are generated from the HLTPs, effectively physicalising the tests and providing detailed instructions for their execution. (LLTSs may take the form of automated test scripts, where applicable.)

## 4.3 Progressive Integration

In common with all large and/or complex IT solutions, it would be unrealistic to expect to successfully integrate the very great number of individual components making up HNG-X, to form a coherent overall solution, all in a single step. Such attempts, without exception, result in chaos, with multiple failures competing for the tester's attention, disguising each other's symptoms, disrupting the flow of the tests, cross-corrupting the data integrity required by the tests, and making diagnosis extremely difficult. A progressive approach is called for.

This entails a series of integration steps, each building on the added stability provided by the previous one. At each step it is important to have a solid, stable foundation to start from. This is provided by verification/validation activity conducted at that level of maturity. The integration step then takes the 'proven' elements and exercises them in combination to integrate them to the next level, ready for further verification/validation.

These verification/validation activities, and the progressive integration steps, are arranged in a series of testing stages. There are four universally recognised levels of integration – Component Level, System Level, Solution Level (where the solution involves multiple systems), and Release Level.

The starting point is the individual Component (the smallest meaningful deliverable, of whatever type, forming a part of the target runtime state). These must first be verified before they can be successfully integrated together to form small sub-assemblies, which in turn must be verified. These sub-assemblies can then be integrated to the next level (System Level) and validated to confirm they are conforming to the relevant system requirements. Again this then forms the appropriate stable foundation to integrate to the next level (Solution Level), bringing all the co-operating systems together and validating them as an overall solution.

Finally (at the Release Level) the solution can be integrated with its deployment mechanisms and the target Live runtime environment, to prepare it for Go-Live. (The deployment mechanisms should not be considered as incidental. They are an important aspect of the system design, and whether implemented in an automated fashion, or by a manual process, form a vital element for successful deployment.)

The stages of testing required for HNG-X, shown in relation to these 4 levels of integration and verification/validation, can be seen in more detail at 4.4 below.

## 4.4 Stages & Types of Testing & Integration

### 4.4.1 Existing Horizon Testing Stages

The testing stages currently used for Horizon are as follows:

- Unit Test
  - Code Review (CR)
  - Module Test (MT)
  - Link Test (LT)
- Platform Configuration Test (PCT)
- Extended Link Test (ELT)
- Interface Validation (IV)
- System Validation & Integration (SV&I)
  - Business Integration Test (BIT)
  - Direct Interface Test (DIT)
  - Volume Test (VT)

  - Integrity Test (IT)
  - Security Testing (ST)
- Release Validation (RV)
  - Migration Test
  - Live System Test (LST)
- Post Office Ltd Testing
  - Accreditation
  - End to End Testing (E2E)
  - Acceptance (UAT)
  - Field Trial
  - Pre-Pilot

Historically, through force of circumstance, the early stages of testing have become somewhat lightweight, whilst the following integration stages compensate for this, being commensurately heavyweight. This lifecycle imbalance comes at a considerable price in terms of time, effort and cost.

Further, the completely separate testing conducted by POA and by Post Office Ltd (the Customer) involves considerable duplication of effort, both essentially setting about the task of planning, preparing, and running a series of lengthy thread based tests scenarios to confirm the end to end data flows through the solution.

Also, the dislocation between POA's Migration testing and Post Office Ltd's End to End testing is unfortunate. Ideally to test the migration effectively it is necessary to run significant functional tests (such as the End to End tests) and in order to test the end to end data integrity is is preferable to use migrated data. Clearly there is significant synergy here that could be exploited to good effect.

The following schematic illustrates this position:

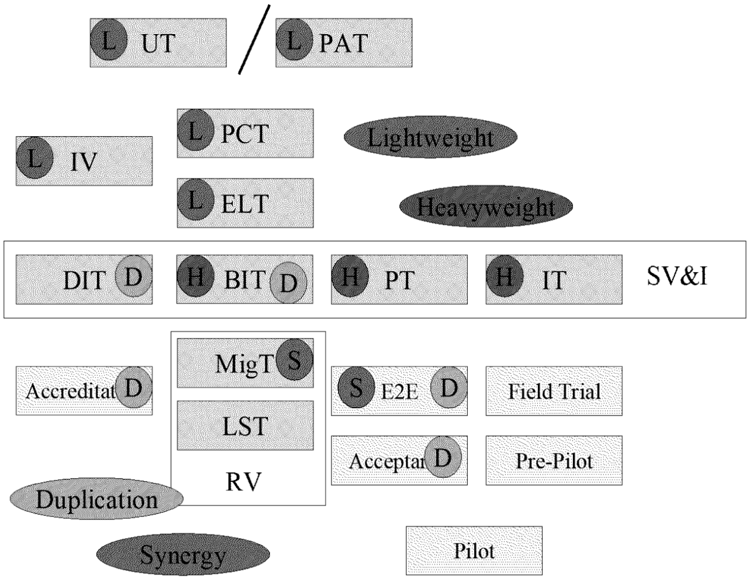


**Figure 14 : Existing Horizon Testing Lifecycle**

### 4.4.2 Addressing the Issues

As HNG-X involves a substantial re-engineering of the Horizon solution, it offers the ideal opportunity to redress this expensive imbalance in the testing lifecycle, applying rigorous verification methods to exploit the benefits of the O-O based development methods. It also offers the opportunity to attack the areas of duplication and to exploit the areas of synergy, merging many of the existing testing stages together and working more collaboratively.

#### 4.4.2.1 Component Level

The existing rather lightweight UT activities are replaced by a new much more rigorous regime. Code Review (CR) will be retained, but given a higher profile than has been common for Horizon, seeking to trap code defects at the earliest opportunity. Adopting the usual industry terminology for component developments, Module Test (MT) and Link Test (LT) are replaced by Component Test (CT) and Component Integration Test (CIT). The new terminology is deliberate, to emphasise the fact that this must be a very different activity to that currently performed. The important characteristics to emphasise are:

- Conducted within development (integrated project teams)
- Exhaustive coverage, made possible by strictly limiting the scope
- Object-Based (Encapsulation)
- Generic basis of testing (synthetic data sensitising full breadth of stimuli)
- Verify against Design
- Include Relevant non-functional requirements (NFRs)
- Comprehensive Regression Packs (insulate wider solution from localised changes)
- Extensive use of Stubs, Harnesses, etc. to enable testing in isolation

**Component Testing (CT):**

Individual component "Z" tested in isolation
Stubs used at each interface

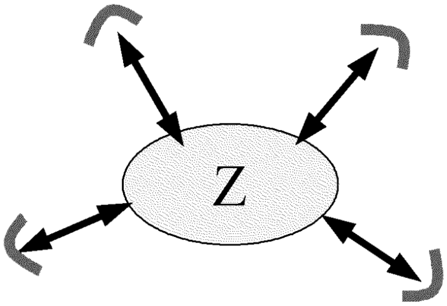


**Figure 15 : Component Test Context**

**Component Integration Test (CIT):**

Component "z" integrated and re-verified together with immediate neighbours, but no further.

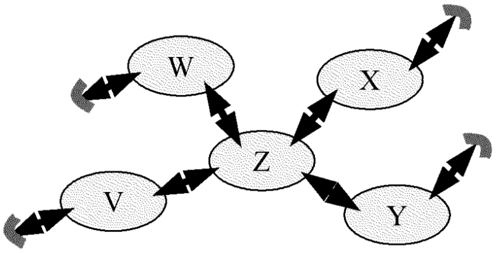


**Figure 16 : Component Integration Test Context**

### 4.4.2.2 System level

The existing rather lightweight PCT, ELT, and IV stages are replaced by a single refocused and more robust System Test stage. The important characteristics to emphasise are:

- Conducted within development (integrated project teams)
- Detailed coverage, made possible by strictly limiting the scope
- Thread based tests (scenarios) integrate components into discrete systems
- Largely synthetic data to improve sensitisation of tests
- Validates  against system requirements
- Integrates applications with supporting infrastructure elements
- Verifies the system interfaces against the interface specifications
- Includes relevant non-functional requirements (NFRs)
- Sample Regression Packs (insulate wider solution from localised changes)
- Some continuing use of Stubs, Harnesses, etc. to enable testing in isolation and to avoid schedule impact through interdependence

### 4.4.2.3 Solution Level

Retain the existing SV&I nomenclature – Solution Validation and Integration. Remove compensation for earlier lightweight stages (no longer applies) and refocus attention on gradual integration of whole solution. The important characteristics to emphasise are:

- Conducted within independent testing unit
- Broad and shallow coverage - shallow here does not imply weak or superficial, but that at this level of validation the test coverage is not intended to be exhaustive, trying to cover every logic path in every combination. (Such detailed testing is performed at the component level.) Here important sample paths through the solution are used to confirm it is properly integrated and that it delivers the overall requirements.
- Thread based tests (scenarios) integrate systems into whole solution

- Largely system generated data to confirm data flow integrity
- Validates against business / operational requirements
- Integrates applications with supporting infrastructure elements
- Validates Systems Management elements
- Starts to introduce migrated data
- Formally Validates the external interfaces against the interface specifications
- Increasing level of collaborative working
- Includes relevant non-functional requirements (NFRs)
- Incorporates Integrity Tests (resilience/recovery)
- Incorporates Performance Tests (load, volume, stress)
- Threads incorporate all BTs and BSTPs to pre-empt acceptance
- Tests planned and prepared for subsequent reuse in RV (i.e. cover both POA and Post Office Ltd objectives and so encompass all the goals that E2E previously had) though at this stage not operating beyond the HNG-X solution boundary (i.e. external systems used only for validating interfaces and not for data flow integrity)
- Sample Regression Packs (insulate wider solution from localised changes)
- Use of Stubs, Harnesses, etc. removed except in support of interface testing and performance testing

Note - For HNG-X, with the development of entirely new layers, and extensive changes to the underlying infrastructure systems, it will not be practicable to operate Integrity Tests or Performance Tests to a sufficient extent from within SV&I (the integrity tests will be too intrusive and disruptive, and the performance tests will need to be operated at very large scale). These will therefore operate as discrete testing stages for HNG-X.

#### 4.4.2.4 Release Level

The RV nomenclature is retained – Release Validation. However, the scope is expanded to also encompass all the earlier Post Office Ltd Testing stages – Accreditation, E2E, Field-Trial, Pre-Pilot testing. This is achieved by combining the POA and Post Office Ltd test objectives (see the description of Objective Driven Testing at 4.1 above) into a single stream of scenario based tests, run interleaved with Migration tests/rehearsals, and so including all aspects of Acceptance testing (to satisfy the Acceptance Criteria specified as to be met by testing). The important characteristics to emphasise are:

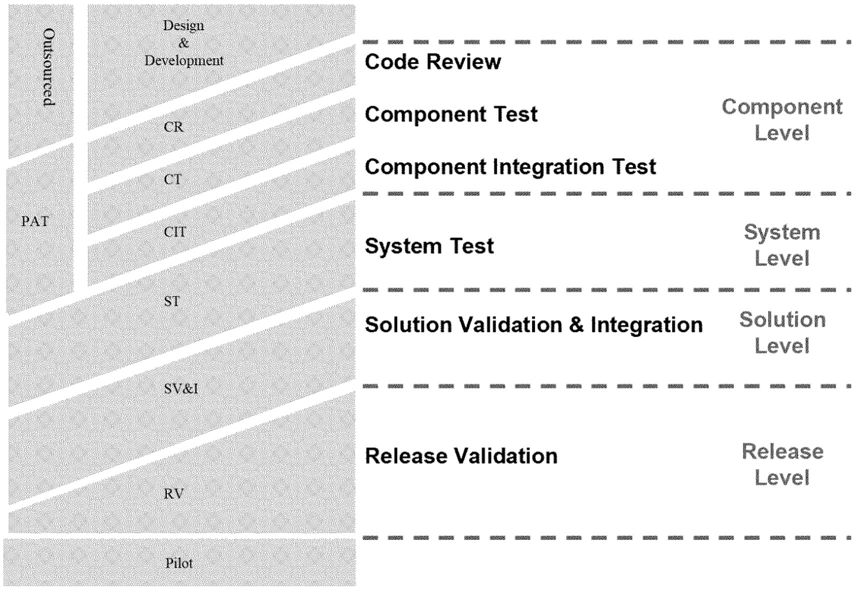
- Conducted within independent testing unit
- Broad and shallow coverage of business functionality (see note on equivalent item under 4.4.2.3 above), with full rehearsal of migration and deployment processes
- Thread based tests (scenarios) interleaved with Migration tests/rehearsals

- Reuses test materials prepared for SV&I
- Incorporates all BTs and BSTPs and so achieves Acceptance testing goals
- Entirely system generated data to confirm data flow integrity
- Validates against business / operational requirements (which for the migration element also involves confirming non-regression from the previous baseline)
- Integrates Solution with deployment mechanisms, including migration
- Further validates Systems Management elements
- Reuses interface validation materials toward achieving Accreditation. (It is important to recognise that if a particular area of Accreditation demands multiple re-runs, or the synchronisation it requires with the third party cannot be organised flexibly (perhaps having to be 'booked' a long time in advance) then this will clash with scheduling the rest of the testing for the RV area, and so in such cases that piece of Accreditation would need to be separated out from the rest to avoid adverse impact.)
- Very high level of collaborative working
- Sample Regression Packs
- All Stubs, Harnesses, etc. removed

### 4.4.3 HNG-X Testing Stages

So, to summarise, the following stages of testing are required for the HNG-X programme, mapping them onto the recognised levels of integration:

- Proof of Concept
  - Architectural Proof of Concept (APOC)
  - Testing Proof of Concept (TPOC)
- Component Level
  - Code Review (CR)
  - Component Test (CT)
  - Component Integration Test (CIT)
- System Level
  - System Test (ST)
- Solution Level
  - Solution Validation & Integration (SV&I)
- Release Level
  - Release Validation (RV)



The following schematic illustrates how these testing stages will in practice overlap in time (for example, not all components are required to enable one system to enter ST, and not all systems are required to enable integration of systems to commence in SV&I).

**Figure 17 : Testing Stages – Overlap**

Product Acceptance Test (PAT) is retained for third party deliverables, to confirm that such deliverables meet the necessary minimum levels of stability of functional conformance so as not to adversely impact the wider testing activities taking place.

In addition, as already explained, for HNG-X it will be necessary to conduct both Integrity Testing (Recovery & Resilience) and Performance Testing (Load, Volume, and Stress) separately from SV&I, as discrete testing stages. Also, each testing stage requires formal planning and preparation, and as already mentioned there will also be Proof of Concept exercises run at the outset. Performance Modelling will commence alongside the APOC, and continue throughout. Complementary Performance Monitoring will be trialled in RV and then run on an ongoing basis thereafter. The following schematic illustrates:

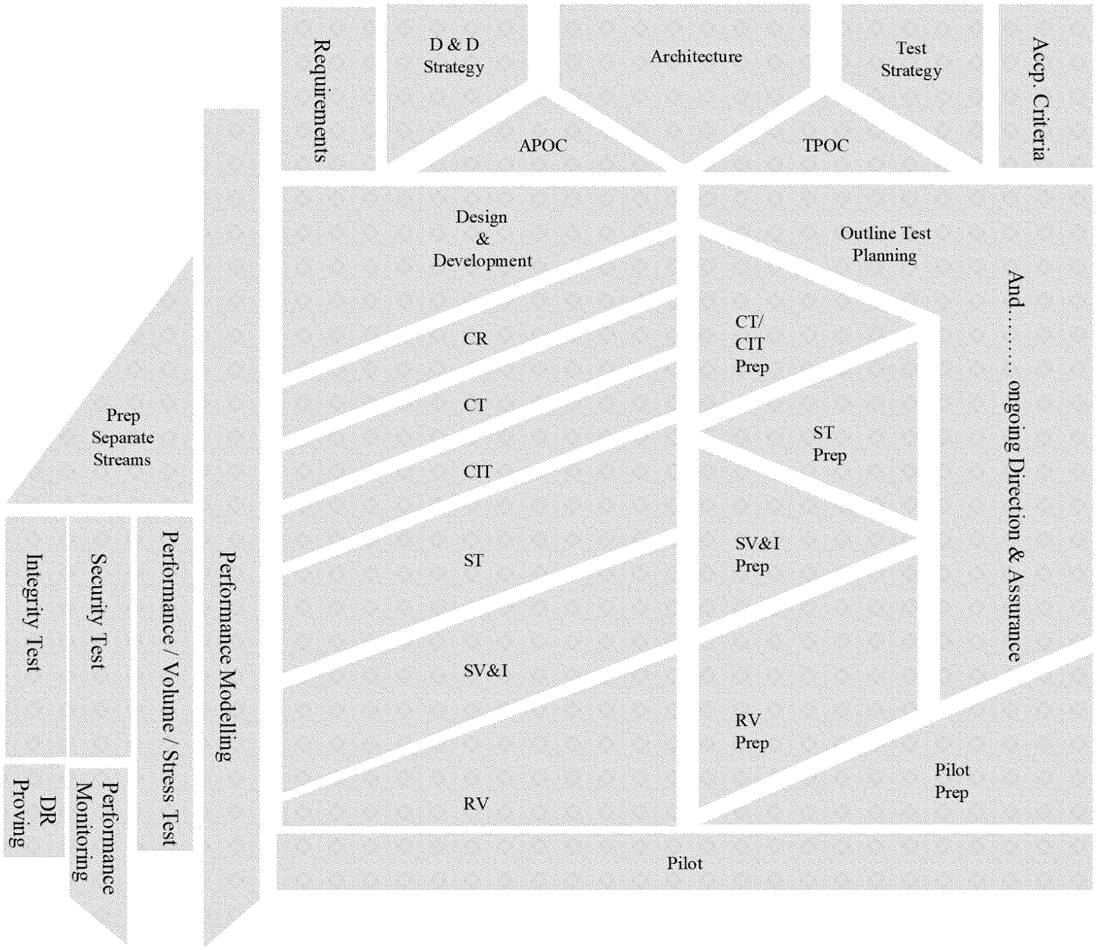


**Figure 18 : HNG-X Testing Lifecycle Overview**

## 4.5 Regression Testing & Retesting

The regression testing and re-testing policy for HNG-X is as follows:

(This provides a brief outline of the regression testing policy for HNG-X, and will be revised in line with the Development Strategy when this becomes available.)

The policy for Regression Testing for HNG-X is that for all change(s) being made to the solution, large or small, simple or complex, a sufficient series of Regression Tests will be run to confirm that (on the balance of probability, and commensurate with the risks involved) those change(s) have not caused the solution as a whole to regress, neither functionally nor non-functionally.

This will be achieved by:

- Strong design separation – this testing approach is predicated on Design & Development adhering to essential component-based development and O-O principles, such that the newly developed areas of HNG-X will be inherently more resistant to invasive change impact. This will be reinforced by Design and Code Reviews, and verified by Component Test (CT), and Component Integration Test (CIT). Change impact will therefore be much more localised than is currently the case with the Horizon Solution.
- CT and CIT test coverage will be exhaustive, and generic in nature, to ensure the design capabilities of each component (and its interfaces) are fully met.
- CT and CIT materials will be retained to form comprehensive packs of Regression Tests at the component level.
- Extended Configuration Management (CM) and Impact Analysis facilities providing change impact details to allow specific changed components, and their neighbouring components, to be targeted for testing
- Related areas of the solution will similarly be identified as requiring specific regression testing at the component level
- The change documents and related sources will be subject to test analysis in the normal fashion, adopting Objective Driven Testing (ODT) principles to determine the basis of risk and so the relative priorities for testing each affected component and neighbouring components, and other areas likely to be affected as indicated by Impact Analysis.
- More general regression testing to confirm the continuing operation of the system(s) concerned and the overall solution will take the form of sample regression tests, selected on the basis of their priority rating(s)
- In general, the smaller, simpler, and more localised the changes are, then the less regression testing will be required, and conversely the larger, more complex, and more invasive the changes are, then the more regression testing will be required.

TBC when Development Strategy becomes available

## 4.6 Flexible Implementation

It has been recognised that the design of many of the existing threads (scenarios) used for Horizon testing (particularly in BIT) leads to unfortunate inflexibility in the manner in which they must be run. Since many of them were created long before the Objective Driven Testing principles were introduced into Horizon testing, they do not benefit from any objective basis of prioritisation. The result is that the scenarios (which are run iteratively as a series of test cycles) often need to be run through to (near) completion in order to achieve desired (high priority) coverage.

An important consideration for the design of threads for HNG-X will be to avoid such inflexibility, and to exploit the objective prioritisation provided by Objective Driven Testing. In this respect, high priority test objectives will, wherever practicable, be built into the early portions of test threads. This will allow for a much more flexible and pragmatic management of test cycles, such that a cycle may be terminated early should the need arise, to allow rapid iteration/re-testing, to 'hit' a high proportion of the highest priority test objectives quickly before diverting effort onto completing perhaps less important testing.

Note – it is a requirement of this approach that the mechanisms necessary to be able to quickly and easily return to test start points (or other checkpoint positions) in order to restart a cycle (or to rerun a particular portion of a thread) will be developed.

## 4.7 Progress by Achievement

A requirement of this approach is to take every effort to prevent 'leakage' of defects from stage to stage. Leakage may occur either by inadequate rigour in planned testing (i.e. insufficient or weak coverage), or by prematurely terminating a stage of testing. This is important because the approach is predicated on rebalancing the testing lifecycle, to invest heavily in earlier testing stages, and so NOT to compensate for poor coverage in later stages. It follows then that the later stages (and ultimately the quality of the deployed solution) relies on the fact that defects will on the whole be trapped at the correct stage. The corollary to this is that if a stage is arbitrarily terminated before it has achieved the test coverage it is planned to achieve, then there will remain a residual (untested) risk. The earlier in the lifecycle this takes place, then the greater the bow wave it will cause and the faster it will escalate.

To combat this tendency, this approach further exploits the benefits of Objective Driven Testing by requiring each stage of testing (and each subordinate area within the stage where applicable) to have Quality Gates defined in advance, stated in terms of the test objectives to be achieved.

In this way highly objective Entry & Exit Criteria (and Suspension & Resumption Criteria) can be specified to realistically control testing progress from stage to stage, on the basis of the test objectives that have been achieved rather than the number of cycles run or some arbitrary planning dates/deadline being reached.

In addition, should a genuine need arise to terminate a stage before these have all been met, then the residual risk (in terms of outstanding prioritised test objectives) will be clearly visible and can be managed accordingly.

TBC when the Development Strategy becomes available

## 4.8 Horizon Based Areas

As explained previously, Horizon will persist (and so will need to continue to be maintained in Production) until HNG-X rollout is completed. The existing processes in use on Horizon have been developed and refined over many years to best suit the peculiar circumstances of the Horizon solution, and these will continue to be used for all Horizon work that may be required in parallel with the HNG-X programme.

Horizon also persists within the target HNG-X solution (the Host systems, many of the Agents, and certain areas of infrastructure and Systems Management). It is likely that the work involved in these areas, albeit within the HNG-X programme, will draw upon common (shared) Horizon based resources. It is not really practical therefore to adopt different processes for these HNG-X areas than will be in use for Horizon. So, to avoid the clash, HNG-X will adopt the existing Horizon processes for all such areas, for all verification, validation, and integration activities up to and including the System Level. The two process streams will then be fused at the point where the systems concerned enter SV&I, where the HNG-X processes will prevail. (This will not cause a clash in process as this testing stage (and all subsequent ones) will be performed outside development, in the independent testing unit.)

Just one exception is imposed by this approach – the Horizon processes will be slightly extended to include creation of regression testing packs for Unit (UT) Test and Extended Link Test (ELT). This is necessary to complete the layer of insulation to protect the wider solution from localised changes.

## 4.9 Newly Developed Areas

The newly developed areas for HNG-X can be divided into two broad classification – those developed in-house, and those outsourced, as follows below:

### 4.9.1 Outsourced Developments

TBC when the Development Strategy becomes available (a brief outline of the approach for outsourced developments is given at section 2 above).

### 4.9.2 In-House Developments

TBC when the Development Strategy becomes available (a brief outline of the approach for in-house developments is given at section 2 above).

### 4.9.3 Iterative Lifecycles and Integrated Mixed-Discipline Project Teams

HNG-X, unlike HNG previously, does not intend to formally adopt the RUP framework for iterative development. It will however utilise iterative development techniques as and where they may be of benefit. Depending on the circumstances the particular brand of iterative development may vary.

For the development of the GUI area, where development would benefit significantly from continuous engagement with Post Office Ltd, a near full-cycle iterative technique has been proposed (i.e. all except actual deployment).

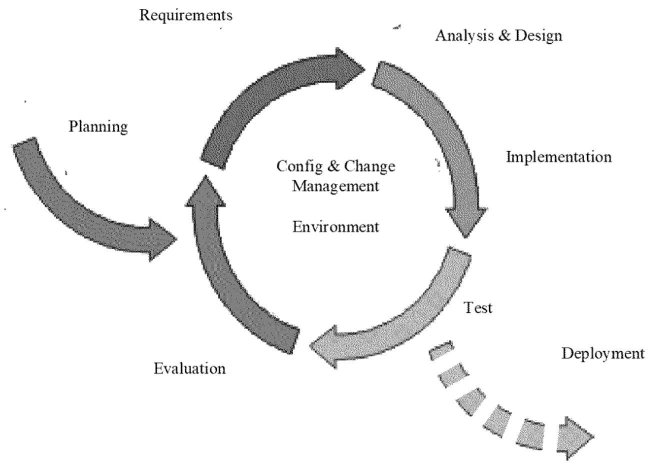


**Figure 20 : Fully Iterative Development Lifecycle**

For the development of the Web Services Layer, which is intended to be an almost 'out-of-the-box' implementation for the Interstage framework portion, there is much less scope for iterative development, although establishing the optimum configuration is likely to be an iterative process of trial/error/refinement.

For the development of the Branch Database, the only iterative element is likely to be refining the design for optimum performance, and subsequent performance tuning.

This all has surprisingly little impact on the strategic approach for testing – testing is always an iterative process, whether as part of a pure Waterfall development, V-Diagram, W-Diagram, Full or Part Iterative development.

The following schematic illustrates the underlying iterative testing process for all development lifecycles:

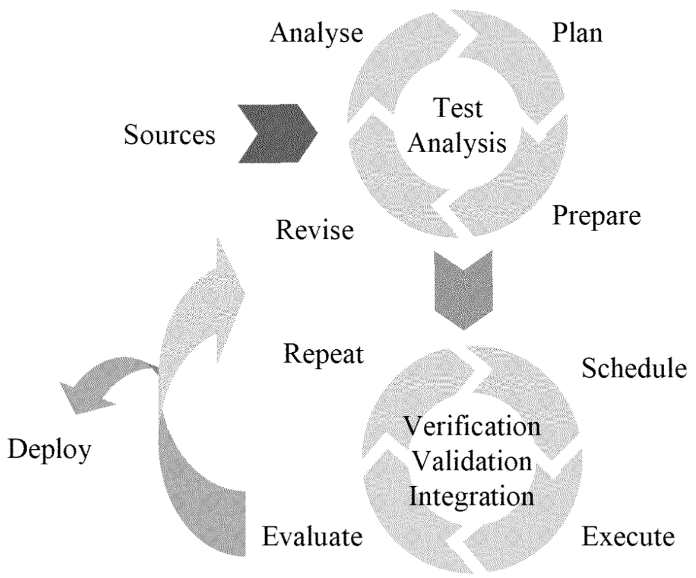


**Figure 21 : Testing Lifecycle – Always Iterative**

Where it DOES affect testing is in the implementation and management of it. Iterative development processes always benefit from being organised as small close-working project teams, working together across the whole of the relevant portion of the development lifecycle (including testing). For Fully iterative developments this would apply to the entire lifecycle. For Part-Iterative developments, such as is more likely for HNG-X, the relevant portion of the lifecycle is that portion which behaves iteratively. So, from a testing and integration perspective, it means those areas of Testing that are performed within the Development zone of responsibility (i.e. the Component and System Levels) and not the areas handled by the independent testing unit (i.e. the Solution and Release Levels).

So, for HNG-X, this approach requires CR, CT, CIT and ST to be conducted within Development, by a mixed-discipline project which encompasses appropriate testing expertise.

## 4.10 Parallel Testing

The HNG-X solution presents a number of unusual characteristics, which when taken in combination offer the opportunity adopt an extremely cost effective testing technique known as Parallel Testing, and HNG-X intends to exploit this opportunity.

The key characteristics of the solution relevant here are:

- The HNG-X programme is predominantly a re-engineering exercise, and for the vast majority of the functionality it is seeking to achieve ‘Business Equivalence’ (i.e. very little new or changed business functionality involved).

- The external interfaces remain entirely unchanged.
- The majority of the external interfaces are file based and batch driven, and the Host systems responsible for generating these interface files also remain unchanged.
- Although the transaction volumes processed by the systems are very large, and the system data transformations and relationships are complex, this results in a relatively low volume of interface data.
- All the interface data is determined as a direct result of the content of the central transaction store (the Riposte Messages in the Correspondence Servers for Horizon, and the tables/rows of the Branch Database for HNG-X.
- There will be a clearly defined migration mechanism to transform the Riposte Messages into equivalent transaction information on the Branch Database
- Conducting equivalent business on both Horizon and HNG-X will result in equivalent interface records being produced on both.

So, simply by running a parallel test on the new HNG-X solution (parallel to the live operation of Horizon in terms of the data present, the transactions conducted, and the interface files generated), then the enormous complexity of the internal processing involved in the two solutions can be reduced to the net resultant interface files produced. The bottom line is that it dramatically reduces the test planning and test checking required to confirm overall non-regression of the solution.

In this case then, it would involve:

- Selecting a well delineated period of Live operation of the Horizon solution. Using the EOD position on a given business day is a convenient method – effectively a business day's worth of transactions. If one of the four correspondence servers is selected, approximately 25% of a daily workload, providing a highly representative spread of transaction types. Alternatively specific branches could be targeted, though naturally the smaller the sample dataset the smaller the range of different business circumstances would included in the test.
- Take a backup of the selected correspondence server(s).
- Allow all that night's batch processing to proceed, producing the interface files corresponding to those transactions, and take a backup of them also.
- Transfer the backups into a test environment (special authorisation would be required to make use of this Live data for testing purposes).
- Migrate the selected transaction data onto the Branch Database of an HNG-X test environment, and run that night's batch processes.

- Secure the resulting interface files, and compare them with those produced by the Horizon Solution. Certain physical differences would be expected (such as time stamp differences and the random chunking of the TPS files) but in business terms they should be identical.

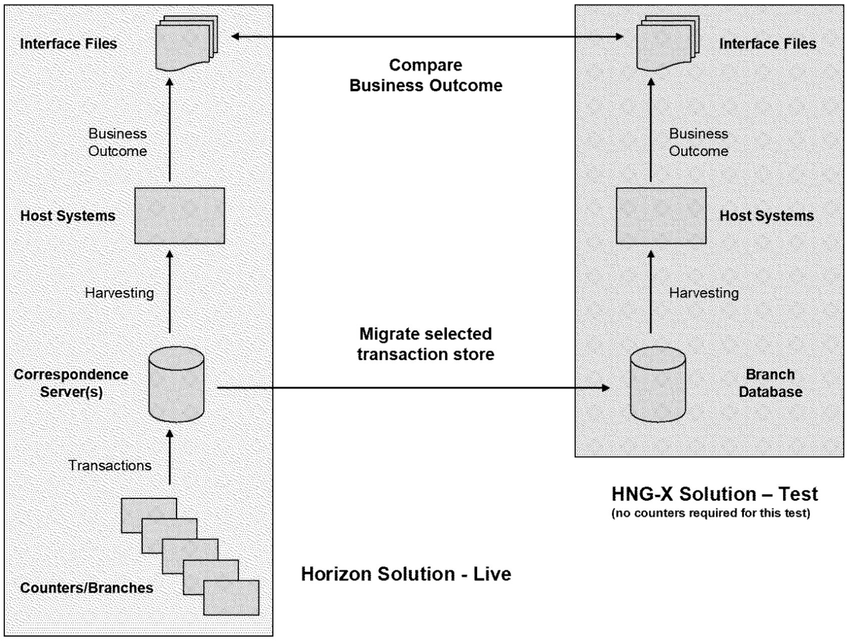


Figure 22 : Parallel Testing for HNG-X

## 4.11 Collaborative Working

Fundamental to this strategic approach is the adoption of fully collaborative working between POA and Post Office Ltd for broad range activities spanning every aspect of testing and integration for the HNG-X programme. This extends well beyond simple witnessing of POA tests by Post Office Ltd, and even joint teams executing certain tests together. Essential to it is a lifecycle long relationship to jointly:

- set the strategic direction (this document);
- engage with the requirements analysis activities to ensure testability and to commence the high level test analysis;
- work with the requirements analysis area to help produce meaningful and practical acceptance criteria in the form of test scenarios (BTs and BSTPs);
- establish a combined set of test objectives covering all the needs of both parties, engaging with both the business and operations to prioritise them on the basis of business and operational risk;

- iteratively expand the catalogue of test objectives as the project lifecycle progresses and additional information emerges, such as by engaging with the system designers to ensure testability of all components, including setting the testability framework, and engaging with requirements and design personnel in assessing changes;
- periodically reassess the basis of risk, consulting with the appropriate business and operational representatives to adjust the related priorities as necessary;
- continue to monitor and adjust the strategic direction and gain assurance throughout the programme;
- plan and prepare test materials for shared use and re-use;
- eliminate duplicated effort, exploit synergy and merge separate testing activities together wherever practicable to streamline the overall testing lifecycle;
- execute some areas within the SV&I stage, including formal validation of the external interfaces;
- execute almost all areas of the RV stage, embracing all previous objectives of the existing Post Office Ltd Testing activities, including Accreditation, Acceptance Test, and Pre-Pilot Test;
- report testing progress, including the contribution toward Acceptance

The intended implementation for HNG-X is to establish a Core Team, comprising roughly 50% POA personnel and 50 % Post Office Ltd personnel. This CORE team is initially involved in setting the strategic direction for HNG-X testing and integration (i.e. producing this document), and then working with the Requirements Analysis areas (for both business and operational requirements), following the principles of Object Driven Testing as described in outline at section 2.3 above, and in more detail at section 4.1 above.

This team would comprise exclusively highly experienced senior testing personnel, and would form a line of continuity throughout the whole lifecycle of the programme, setting, monitoring, and refining the strategic approach, assisting and advising the testing teams, and continuously gaining assurance.

Additional Post Office Ltd resources would later be injected into the testing teams, for the test planning and preparation activities, and again for RV test execution (including a contingent of End Users to assist in validating Procedures, Instructions, and Training materials, and to help man the Model Office environment.

The roles and responsibilities involved are indicated at section 6.0 below.

## 4.12 Defect Management

This area will be actively developed as part of the Testing Proof of Concept (TPOC) exercise. However, at a strategic level the key points are:

- All defects identified by formal testing will be recorded
- Where defects are found in component level testing, conducted by developers, they need not initially be raised as defects in the central defect management system, as it is most likely the same developer who will have to diagnose/fix the bug, and using the defect management tool would impose an unnecessary overhead. However, they will be noted in a defect log for inclusion in later metrics (they may for example help identify issues in a particular area of the design, or leakage occurring through inadequate Code Reviews).
- Where defects are found in later stages of testing, they will be recorded in the central defect management system, in all cases.
- Defects will be related to the test stage/plan/script/step/environment concerned, which through traceability will in turn relate to the test objective(s) and requirement(s)/design(s) concerned. After diagnosis this will also be related to system/component.
- Defects will be assigned priorities/urgencies/importance on the basis of the priority of the related test objective(s), the technical criticality of the system/component, the potential business impact, the potential testing impact (cost/schedule/coverage/dependency), any acceptance or release authorisation consequences, and also perhaps the defect density in that area (if has become an issue).
- Outstanding defects will be formally reviewed on a regular basis (frequency driven by need – monthly, weekly, daily) by a Problem Review Forum (PRF) to assess/re-assess priority and likely impact. The PRF will have business, operations, testing, design and development representation, working collaboratively.
- Defect rates will be actively tracked as an integral part of test management

TBC when the tooling strategy has been determined (comes out of the TPOC exercise)

## 4.13 Environments, Tools & Automation

This is an area will be actively developed as part of the Testing Proof of Concept exercise. However, at a strategic level the key points are:

- Environments
  - HNG-X will have three main sets of test equipment available.
    - Existing test estate currently used for Horizon, increasingly becoming available as the Horizon testing usage diminishes over time (this is not planned to be retained beyond HNG-X)

        - New HNG-X data centre equipment to be installed well in advance at the "Active" site (this will of course cease to be available for testing use at the time it is required to start preparing for Data Centre Migration)
        - New HNG-X data centre equipment to be installed well in advance at the "Standby" site (this will remain for testing use throughout, except when DR is invoked or rehearsed)
    - Mechanisms will be developed to flexibly deploy the equipment at the data centres, enabling:
        - Multiple parallel environments capable of supporting multiple parallel testing streams
        - Rapid and reliable reconfiguration and redeployment to form varying numbers of different shaped environments as the testing lifecycle unfolds
        - Easy taking of 'checkpoint' positions before, during, and after testing cycles (configuration, codeset baseline, data)
        - Rapid and reliable resetting of environments to nominated 'checkpoint' position
        - Separate data storage provision for Live and Test
        - Capability to switch (under appropriate controls) between Live and Test datasets/networks/keys (to support pre-production testing and parallel testing methods)
        - Failsafe isolation of Standby site equipment and network which is in use for testing (external interfaces, Active site synchronisation, etc.)
        - Practicable remote management facilities – configure, install, regress, time & date manipulation, systems management, batch scheduling, etc.
    - Range of different environment types configurable
        - Full DR
        - Large-scale, Live shaped (e.g. for Performance, Stress)
        - Medium-scale, Live shaped (e.g. Resilience, Migration)
        - Small-scale, Representative shape (e.g. SV&I)
        - Small-scale, Synthetic (e.g. ST, Interfaces)
    - Component level testing will be performed on development equipment
- Tools
    - There is no drive on HNG-X to replace tooling across the programme in support of RUP, as was the case previously for HNG – in general it should be assumed that the existing tools will continue to be used unless the nature of the task concerned is new or dramatically changes, or there is some other compelling reason to adopt a different toolset.
        - PVCS for CM, but metamodel extended to provide necessary structure in support of O-O etc, and to provide impact analysis
        - TestDirector for test planning and test management

        - PEAK for defect management
        - LoadRunner for performance testing
        - Win Runner for functional test automation
        - New tools for Component level testing (JTest, JUnit, etc.)

      The usage/requirement for each tool will decided as part of the TPOC exercise, ensuring that they join up, an the associated processes are properly aligned.
    - There will be a need to adopt consistent working practices in the areas employing collaborative working. Typically this will mean standardising on the tools used by Fujitsu Services testing. (For example, it would be sensible to use a single defect management tool for all testing.)
- Automation
    - The testing automation policy for HNG-X can be summarised as: automated testing techniques will be adopted wherever and whenever it provides a clear benefit. The adoption must therefore be pragmatic, embracing automation where it can be readily applied, and avoiding it where costs become prohibitive. The detailed test automation strategy will need to be developed, but it is clear that the primary focus should be in the early verification stages (these are small, self contained, and have fewer data dependencies to contend with, but require exhaustive coverage, making them ideal candidates for automation). Automation of the later stages of testing should be approach more critically, adopting automated scripts only where the benefit is clear, and where the cost of future maintenance of scripts will not escalate. (Sample coverage to support minimum regression testing would seem appropriate.)

TBC when the environment and automation strategy has been determined (comes out of the TPOC exercise)

## 4.14 Metrics

This is an area which must be closely aligned with the associated tools and automation (see 4.12 above). In order to avoid undue overheads, it is important to generate any metrics that are required as an integral part of the underlying process and wherever possible do so automatically and transparently. Their precise definition is therefore inextricably entwined with that of the tools and automation being used. However, at a strategic level, the metrics that must be collected are:

- Time and Effort against Plan
    - Manpower resource expenditure by activity/task/skill – informs current budget position and future manpower estimating
    - Activity/Task progress tracking against project plan – informs current schedule position and future timescale estimating

    - The standard POA planning and time recording processes will be followed
- Testing Progress / Product Quality
    - Test Objectives derived, by nature/priority
    - Test Plans/Scripts/Steps produced, by stage/priority
    - Test Plans/Scripts/Steps executed – Success/Fail, by stage/priority
    - Test Coverage Achieved – by Test Objectives covered
    - Defects - Raised, Closed, False, Outstanding – by priority and by component/area, over time, traceable to Test Plan/Script/Step and to Test Objective
    - Handovers Processed – by component/area, over time
- Process Improvement
    - Root Cause Analysis and Defect Density – by component/area
    - Re-Handovers – by component/area
    - Defect Leakage – by component/area and by test stage
    - PONC (Price of non-conformance)

TBC when the environment and automation strategy has been determined (comes out of the TPOC exercise)

## 4.15 Process Improvement

Process improvement will be an integral part of the HNG-X mode of working for all testing and integration activities. As such there will be a post stage review for each stage, and a post release review for each release, specifically to examine the performance and achievement of the testing and integration activities in that stage/release, highlighting any potential process deficiencies and recommending corrective action/process improvement accordingly.

# 5.0 Testing Coverage

This section describes the intended testing coverage for each of the target systems under test, which together comprise the HNG-X solution. This section cannot be populated at the outset, as a significant level of test analysis work is first required. (At this time a bare skeleton of the areas to be considered can be found at section 2.1.4 above.)

When the initial Test Objectives have been determined (from the System Requirements, etc.) and have been prioritised, then an outline feature set will be documented.

Later, when the high level test analysis activities have been completed, and prior to producing the HLTPs (for each stage) further detailed information will be available to expand these entries.

## 5.1 Features to be Tested

TBC initially at an outline level following prioritisation of the Test Objectives arising from analysis of the System Requirements, and later in more detail following completion of the high level test analysis activity, where details of the relevant features will emerge (prior to completion of the HLTPs)
Ensure that ‘router testing’ is explicitly included in this list.

## 5.2 Features NOT to be Tested

TBC initially at an outline level following prioritisation of the Test Objectives arising from analysis of the System Requirements, and later in more detail following completion of the high level test analysis activity, where details of the relevant features will emerge (prior to completion of the HLTPs)

# 6.0 People, Skills & Responsibilities

The following table indicates the primary roles and responsibilities applicable to each of the major activities:

| **Key:**<br>**P = POL, F = Fujistu Services, J = Jointly**<br>**(+3P = together with the 3rd party concerned)**<br>**Activity** | **Owns** | **Directs/Guides** | **Manages Work** | **Agree/Sign Off** | **Main Resource** | **Supports** |
|---|---|---|---|---|---|---|
| Programme Test Strategy | J | J | F | J | J | J |
| Involvement in Business Requirements/Accp Criteria | P | J | P | J | J | J |
| Involvement in Operations Requirements/Accp Criteria | F | J | F | J | J | J |
| Derive Test Objectives | J | J | F | J | J | J |
| Risk Assessments – establish Priorities | J | J | F | J | J | J |
| Outline Test Planning | J | J | F | J | J | J |
| CT | F | F | F | F | F | - |
| CIT | F | F | F | F | F | - |
| ST | F | J | F | F | F | P |
| SV&I – Main | F | J | F | F | F | P |
| SV&I – External Interfaces | F | J | P | J+3P | F+3P | P |
| Performance Testing | F | J | F | P | F | P |
| Integrity Testing | F | J | F | J | F | P |
| Security Testing | F | J | F | J | F | P |
| DR Proving / Rehearsal | F | J | F | P | F | P |
| RV – Migration aspects | F | J | F | P | F | P |
| RV – E2E aspects | J | J | F | J | J | J |

| | | | | | | |
|---|---|---|---|---|---|---|
| RV – Deployment aspects | F | J | F | P | F | P |
| RV – Accreditation aspects | P | J | P | P+3P | J+3P | F |
| RV – Acceptance aspects | P | J | P | P | P | F |
| RAB | P | J | P | P | P | F |
| Pilot | P | J | P | P | P | F |

# 7.0 Assumptions, Dependencies, Risks & Constraints

TBC – this section to be populated initially by abstracting all references to assumptions, dependencies, risks and constraints from the body of the document – next issue

## 7.1 Assumptions

TBC as above

- The primary driver for the HNG-X programme is to reduce the TCO. Whilst it is desirable in so doing to also improve the responsiveness to future change (reduced Time to Market, and reduced cost of delivery) these are not drivers for the HNG-X programme and can only be pursued on a cost neutral basis. Accordingly, this approach is not optimised for future change, but does exploit opportunities to benefit this area where they occur in satisfying the programme drivers. (e.g. stipulating the need for generic and exhaustive component level testing, and the need to retain comprehensive regression test packs, will requires significant investment, which whilst it will be recouped within HNG-X is unlikely to realise significant savings overall, but will benefit future changes.)
- Both POA and Post Office Ltd desire a single, consistent, joined-up testing approach for HNG-X.
- Both POA and Post Office Ltd desire, and are prepared to support and resource, fully collaborative working for the testing of HNG-X, wherever this may result in reduced duplication of effort between the parties, and/or exploit synergy to improve efficiency and/or effectiveness.
- The will be a lengthy Pilot period (about 6 months) where the throughput volumes for the new HNG-X systems will remain low (perhaps 100-300 branches) – this mitigates performance risks at first Go-Live.
- Following Pilot, there will be a gradual migration of branches to the new HNG-X systems (over a period of about 6 months), and so there will be a corresponding gradual ramp-up of workload volumes, which further mitigates performance risks whilst the migration proceeds.
- The Host systems and the external system interfaces employed in Horizon will remain unchanged for HNG-X
- Development work outsourced to a third party (whether local or offshore) will undertake component level testing with at least equivalent rigour to that outlined in this approach for in-house developments, and will provide the same essential testing deliverables, including comprehensive Regression Test packs.
- There will not be a requirement to perform End-to-End Performance Tests involving external systems.

- Preparatory changes (such as relocating the datacentres, installing the branch routers, and installing the additional hardware in the new data centres) will be conducted under the mantle of Horizon changes and so are covered by the existing Horizon test approach.
- The HNG-X solution will adopt an Active/Standby configuration for Disaster Recovery (DR), and the equipment at the Standby site (intended for DR) will ordinarily be available for testing purposes, except when DR is invoked.
- HNG-X will benefit from a gradually increasing ability to exploit targeted testing techniques, with the corresponding ability to perform accurate impact analysis, as the new developments progress.
- A Testing Proof of Concept (TPOC) exercise will be conducted prior to the main development, to develop, pilot and tune any necessary testing processes, tools, and automation methods.

## 7.2 Dependencies

TBC as above

- Post Office Ltd Business Requirements and Fujitsu Services Operational Requirements will be produced at the outset of the programme, to formally identify both the functional and non-functional requirements for Business Equivalence and the reduction in Total Cost of Ownership.
- There will be an opportunity for Post Office Ltd and POA testers, working together, to engage with the corresponding requirements teams and to ensure that the emerging requirements are testable.
- Any Acceptance Criteria specified as to be satisfied by means of testing, will be couched in terms of the test scenarios necessary to demonstrate that they are met, and Post Office Ltd and POA testers working together will assist in articulating these test scenarios as formal Business Threads (BTs) and supporting Business Specified Test Points (BSTPs) to be carried forward directly into later test planning activities.
- Post Office Ltd will provide the necessary business engagement, and POA will provide the necessary operational engagement, in a series of workshops with Post Office Ltd and POA testers working together to conduct a highly granular risk assessment of the derived Test Objectives, and facilitate the prioritisation of those Test Objectives on the basis of such risk assessments.
- The equipment at the Standby site will be organised such that it can be quickly, cheaply, and easily be reconfigured to provide multiple, flexible test environments, which will support the taking of rapid 'checkpoints' and the rapid resetting of the environments to such checkpoints.
- It will be acceptable for Live Data to be used in HNG-X testing (under appropriate strict controls) in order to exploit Parallel Testing methods and so avoid significant time-consuming and expensive regression testing that would otherwise be necessary.

- An Architectural Proof of Concept (APOC) exercise will be conducted prior to the main development, including
    - Commencing performance modelling
    - Proving viability of outline architecture and related technologies
    - Proving intended scalability mechanisms
    - Gaining an indication of likely performance characteristics
- Integrated project teams, with mixed-discipline personnel, will be employed for the development of the new HNG-X areas, such that experienced testing personnel will be involved throughout
- The Unit Test (UT) and Extended Link Test (ELT) areas of Horizon will extend their existing processes to retain and organise their test materials as Regression Test packs, for those systems which will form a part of the HNG-X solution (e.g. the Host and Agent systems).
- The HNG-X development area will adopt component-based development and O-O principles to ensure that a strong design separation is achieved (applies equally to outsourced developments also).
- The HNG-X development area will adopt exhaustive, generic, component level verification methods, in accordance with this strategic approach - Component Test (CT) and Component Integration Test (CIT) – and retain comprehensive Regression Test packs at this level.
- The HNG-X development area will conduct system level validation methods in accordance with this strategic approach – System Test (ST) – for all systems forming a part of the HNG-X solution (i.e. in-house HNG-X development, outsourced HNG-X developments, and Horizon systems redeployed or updated to form a part of HNG-X).
- The existing Configuration Management (CM) system / toolset / processes will be extended appropriately to accommodate the CM requirements of component-based, O-O development and testing, including supporting the necessary relationships to support effective impact analysis, and providing the necessary information to facilitate automatic configuring of hardware platforms and automatic building/installing/configuring of software stacks.
- The POA and Post Office Ltd testing personnel provided for the collaborative working required by this strategic approach, and in particular those forming the CORE testing team, will be:
    - Appropriately skilled, highly experienced testing personnel
    - Empowered to make all necessary testing and test management decisions the role will require of them
    - Be assigned such that continuity throughout the programme will be maintained
    - Be supported by their respective managers in a manner that does not compromise their collaborative working role

## 7.3 Risks

TBC as above

- If the necessary design separation is not achieved or the necessary rigour in component level testing is not applied to exhaustively and generically verify each component, then the subsequent testing stages as described in this strategic approach will be unable to provide the in-depth detailed testing coverage to compensate for this deficiency and so the approach will need to be entirely revised.
- If the personnel involved in the collaborative working areas are not appropriately empowered, then the decision making processes will become protracted, significant levels of rework will resul, and so costs and schedules will be adversely impacted.
- If CM is not extended as intended, then accurate targeted testing will not be possible, resulting either in inappropriate testing coverage (reduced product quality) or very extensive regression testing (increased costs and timescales).
- If the Acceptance Criteria are not produced at the outset, and couched in terms of the necessary test scenarios, it may not be possible to accommodate them collaboratively when planning and engineering the test materials as intended, and so may necessitate the re-separation of Post Office Ltd testing stages, with all the duplication of effort and increased timescales that will involve.

## 7.4 Constraints

TBC as above
No specific constraints identified at this point in time