FUJ00119499

Document Title: TESTING & INTEGRATION STRATEGY

Document Type: Strategy

Abstract:

This document defines the strategic approach to be adopted for the testing and integration of Pathway products for BA/POCL. Its scope extends well beyond the application software to cover the entire system, from unit testing of individual software modules, through integration of software and hardware, to trials of the fully configured operational system. It should be read in conjunction with the General Testing Policy document. It covers the areas of functional conformance, system performance and resilience, end to end architectural integration, system operability, and business integrity. It also provides a framework for the allocation of specific areas of responsibility. It excludes the detailed verification of the many bulk standard proprietary software packages and hardware items (shrink wrapped and off the shelf goods) other than in the context of its specified usage by this system. An overview of the high level test processes to be employed are included as an appendix. For further detail see the separate Testing Processes document.

Status: Approved

Distribution: Pathway Management Team
(and further at their discretion)

Author: S Berry & J Hunt

Approval Authority: Ian Honner
Signature/Date:

Quality Authority: David Groom
Signature/Date:

DOCUMENT HISTORY

Version	Date	Reason
0.1	05/01/96	Initial draft - general framework - issued for review, key Pathway personnel only
0.2	19/01/96	Internal version only - not issued
0.3	02/02/96	Internal version only - not issued
0.4	16/02/96	Final draft - full document - issued for wider review, Pathway personnel across programme
1.0	01/05/96	1st approved issue, following final review

CHANGES FROM LAST ISSUE

Ref.	Change
n/a	Comments applied from final review
A2	Glossary section included
12/13	Verification Centre changes applied (sequence in lifecycle, and limitations)
n/a	Section on Estimates and schedules removed
n/a	Section on Quality and Risks removed
6.2	Changes relating to decisions on Release Strategy included
17	Section on external certification included
n/a	Status changed from draft to approved

CHANGES FORECAST

Change	Target Issue
Changes relating to issue of SLAs	2.0
Changes relating to issue of Acceptance Criteria	2.0
Changes relating to issue of TED	2.0
Changes relating to agreementof final PBFS	2.0
Changes relating to process revisions	2.0

ASSOCIATED DOCUMENTS

Ref.	Version	Date	Title	Source
[1]	1.0	01/05/96	General Testing Policy	Pathway
[2]	1.0	?	Acceptance Criteria	BA/POCL
[4]	1.0	?	Migration Strategy	Pathway
[5]	1.0	?	Implementation Strategy	Pathway
[6]	1.0	?	Technical Environment Description	Pathway
[8]	1.0	?	Pathway Baseline Functional Specification	Pathway
[9]	1.0	?	Risk Response Catalogue	Pathway
[10]	2.0	?	Configuration Management Strategy	Pathway
[11]	1.0	01/05/96	Testing Processes	Pathway

ABBREVIATIONS

Used	Meaning
API	Application Program Interface
BA	Benefit Agency
BOM	Bill of Materials
BRTS	Business Release Test Strategy
c.f.	see, reference, compare (confer)
CM	Configuration Management
EPID	External Physical Interface Definition
HCI	Human Computer Interface
HLTP	High Level Test Plan
I/F	Interface
IT	Integration Test, or sometimes Information Technology
ITS	Integration Test Strategy
LLTS	Low Level Test Script
LT	Live Trial
LTS	Live Trial Strategy
NFR	Non Functional Requirement
OHE	Output Handling Equipment
PAT	Product Acceptance Test
PATS	Product Acceptance Test Strategy
PBFS	Pathway Baseline Functional Specification
PO	Post Office
POCL	Post Office Counters Ltd.
RAD	Rapid Application Development
SLA	Service Level Agreement
ST	System Test
ST	System Test
STS	System Test Strategy
TED	Technical Environment Description
TEP	Test Execution Plan
TIS	Testing and Integration Strategy (this document)
UAT	User Acceptance Trial
UAT	User Acceptance Trial
UATS	User Acceptance Trial Strategy
UT	Unit Test
UTS	Unit Test Strategy
VC	Verification Centre

CONTENTS

APPENDICES

1. INTRODUCTION

1.1 This document defines the overall strategic approach to be adopted for the testing and integration of products by Pathway for BA/POCL, as required in section 1.3 of the General Testing Policy [1]. This document is subordinate to the General Testing Policy [1] and describes how the testing and integration activities will apply the policy.

1.2 At a high level it describes the stages of testing and integration to be carried out, and includes high level processes for each of these stages in Appendix A. More detailed descriptions of these procedures can be found in the Testing Processes document [11].

1.3 This document is at the head of a family of documents. It is intended to be very much a work document, and not shelfware. Implementation is via the production of these detailed subordinate strategy documents which serve as the vehicles to enhance the approach described here and to add further appropriate detail as it becomes available during the development lifecycle. Together they unambiguously define the objectives, scope, coverage, and success criteria for all the testing necessary to meet the Customer agreed Acceptance Criteria [2] for each major release. The following figure maps out these documents and their inter-relationships.

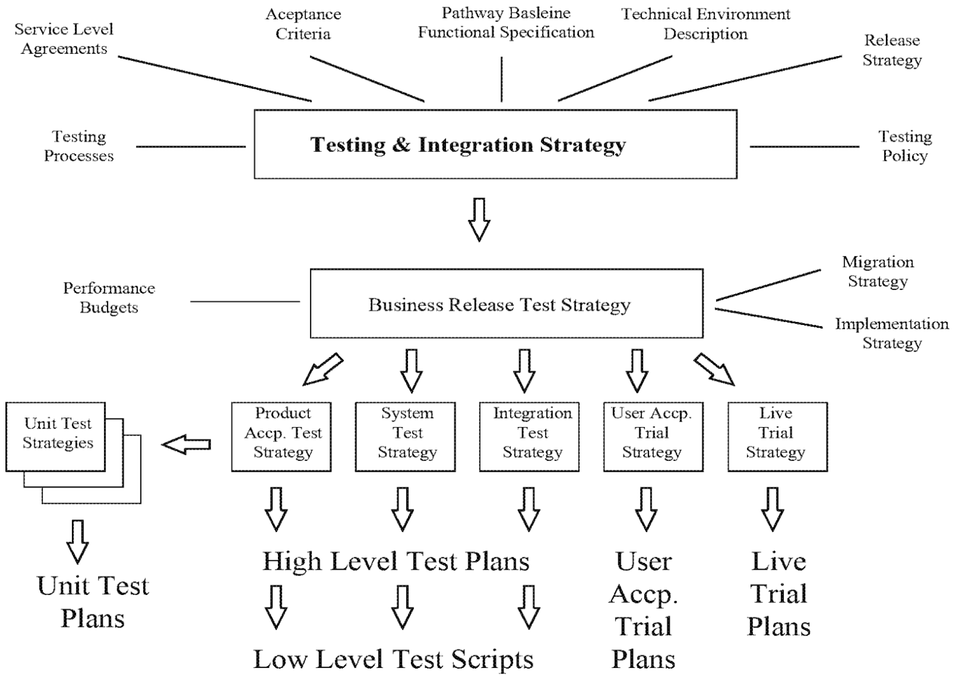


Figure 1 - Family of Documents defining Test Coverage

1.4 The Pathway Baseline Functional Specification [8] is agreed with the Customer within 30 days of the award of contract, and forms the fundamental document driving out the testing that must be considered under this approach. (It is structured so as to serve also as a requirement catalogue for audit and traceability purposes.) Together with the Technical

Environment Description [6], the Customer agreed Acceptance Criteria [2], and the Service Level Agreements (SLAs) these four serve as the principal inputs to high level test planning. Taken in conjunction with the General Testing Policy [1] and the testing processes (c.f. Appendix A and the Testing Processes document [11]) they forge the high level objectives for all testing and integration activities required on this programme.

1.5 The Release Strategy defines the phased release of software into the Live arena (c.f. section 6 - Release Strategy). Each release may have specific attributes peculiar to that release, corresponding with the particular products being released at that time, and related to the Migration Strategy [4] and Implementation Strategy [5] for that release. There are also likely to be specific performance issues for each release which will be specified by the Performance Budgets derived from the TED [6]. These peculiarities will be detailed in the Business Release Test Strategy for that release, as regards any special testing and integration features that may pertain.

1.6 This will then be expanded down into specific testing strategies and High Level Test Plans (HLTPs) to cover each of the stages of testing. These will detail the specific scope, coverage, objectives and success criteria that apply. In addition for each in-house product a Unit Test Strategy will be produced, each in turn giving rise to Unit Test Plans covering both module testing and link testing.

1.7 Finally, when sufficient detail becomes available, the ‘logical’ HLTPs can be translated into ‘physical’ Low Level Test Scripts (LLTSs) and their supporting test data, ready for test execution when the software products become available to run.

1.8 Throughout the test execution period, regular checkpoints will be taken (typically by the taking of physical database dumps and the preserving of associated flatfile and configuration data) at suitable quiescent points during the running of lengthy tests suites. This is necessary to allow tests to be restarted just a little prior to the point of failure on receipt of a fix, or to re-run appropriate segments of a test when regression testing changes to the system. This is particularly important during the System Test and Integration Test stages where the cost of such retesting would otherwise quickly become prohibitive.

1.9 Testing will be managed, through both planning and execution phases, on a strict cost benefit basis. How much testing can be afforded? How little testing can be afforded? The cost of performing tests versus the risk of not performing them. A pragmatic approach to test planning and execution is essential in maintaining the correct balance, cost versus risk. As test planning progresses there will be areas of the system deserving of more scrutiny than others, because of the inherent risks they entail. More time and effort will be allowed here accordingly. Similarly certain areas will be deserving of little or no scrutiny, being of very low risk impact, and so tests will be reduced or discarded here. (Please note that where ‘cost’ benefit’ and ‘risk’ are concerned above, than ‘time’ and ‘schedule’ are contributary elements. Lost time costs, saved time can benefit, and schedule dependencies and late delivery constitute project risks.)

2. MANAGEMENT SUMMARY

This document sits at the head of a family of strategy documents which together serve to unambiguously describe the scope and coverage of testing required for the programme, and in this way they also serve as the means of agreeing this test coverage with all interested parties.

The approach to testing is one of staged, systematic verification, with progressive integration of software and hardware components, culminating in overall system validation of the fully configured system in the Live environment.

The system is subjected to testing against three principal test life-cycles, for Functional Conformance, Architectural Conformance and Business Integrity. These give rise to six stages of testing - Unit Test, Product Acceptance Test, System Test, Integration Test, User Acceptance Trial and Live Trial.

Unit Test (UT) deals with the detailed verification of individual modules and their low level linking to form products. It is performed explicitly for all products developed in-house. An equivalent level of testing is assumed to be performed by the other suppliers.

Product Acceptance Testing (PAT) is performed for each bespoke software product, and is as its name suggests the means of formally accepting a product into the programme, under Configuration Management (CM) control.

System Test (ST) is performed against a full software set and serves to validate the software against the requirement, concentrating on functional conformance. It operates the products in conjunction with each other and follows business threads.

Integration Test (IT) takes the system tested software set, properly configured and installed on the full live target hardware set, according to the CM Bill of Materials (CM-BOM), and including links with external systems, and validates the interworking of the full system, puttting it under load and stress to verify the performance characteristics.

Working from the expanded CM-BOM, now validated by IT, the Verification Centre (VC) builds the agreed baseline system configuration (for the 'counters' sub-systems) ready for rollout.

Rollout is first to the User Acceptance Trial (UAT) which is effectively a continuation of Integration Testing, but performed in a model office environment, and principally operated by the customer to serve as the vehicle for formal acceptance of the system from the Service Provider. Various business support products are brought into play here also, including the Help Desk.

Rollout is then to the Live Trial (LT) which is a limited but actual live operation, with live staff, live data, and the general public involved in real transactions at a number of actual Post Offices - a Full Pilot.

These stages apply to each separate major release of the system, as defined in the Release Strategy. They also apply, where appropriate, in the maintenance of each release. In addition certain products may require a level of external certification for legal reasons. For example the electronic weigh scales used in conjunction with the PCs on the counter top will require certification by the wieghts and measures authority.

Throughout test planning and test execution, testing is conducted by agreement with the programme and the customer, on a cost versus risk basis. No system can ever be error-free. It is not possible to prove that a system works, only that it does not. It is therefore only possible through testing to reduce the risk of error remaining. The cost of this error finding and removal follows a course of increasing cost and diminishing return. The equation is one of cost versus risk. Different systems can abide varying levels of risk.

Each stage of testing is conducted in its own style of test environment, which progressively gets larger, more shareable, less simulated, more live-like and under stricter CM control from stage to stage. This helps to concentrate test activity in more affordable and more appropriate environments and so avoids unnecessary escalation of machine resource costs.

3. SCOPE OF TESTING

3.1 The scope of testing and integration activities encompassed by this document and its subordinate test strategies includes the entire application software system (both 3rd party supplied and in-house developed products) and its integration with the supporting hardware and infrastructure software platform.

3.2 It does not cover specific detailed testing of that platform (which is assumed to be of Assured Status) other than in respect of its support of the software system and in satisfying the SLAs and Non Functional Requirements (NFRs).

3.3 It does not detail the earlier activities of the 'Demonstrator' phase, but rather starts with Unit Test (c.f. section 9), an onward up to Live running in the 'Live Trial'.

3.4 It does not cover the testing activities of the 3rd party suppliers. It is expected that all products supplied will be appropriately tested prior to delivery to ensure that they meet the supplied specifications. However, it is not expected that they will be integrated and proven together, excepting where a supplier is charged with producing interlinking products, where limited link testing is expected.

4. HIGH LEVEL TEST OBJECTIVES

4.1 Specific

For each release, tests are to be engineered to demonstrate the following:

a) each module developed in-house to be compliant with the corresponding module specification and to link correctly with co-operating modules in that product. (Unit Test).
b) each bespoke software product, either developed in-house or provided by a 3rd party supplier, to be compliant with its Product Description/Detailed System Specification. (Product Acceptance Test).
c) software products to operate successfully in conjunction with one another to satisfy the Pathway Baseline Functional Specification [8] and to remain consistent with their Product Descriptions. (System Test).
d) software systems to operate and co-operate successfully together on the target hardware and infrastructure software platform and to interface correctly together and with the BA/POCL systems and other external systems, to satisfy the SLAs and NFRs that apply. This to include verification of Output Handling Equipment (OHE) operation as required by the system, such as card production, use of 'slip printers' at the counters, use of A4 cut sheet printers in the back office, and all associated stationery and materials. (Integration Test).
e) the full operational configuration to operate successfully with user procedures and within a 'laboratory' office environment, and to be accepted by the Customer as fit for service. This to include validation of OHE operation. (User Acceptance Trial).
f) accepted system to successfully support the 'Live Trial' in verifying user procedures, training material, support mechanisms, and help desk procedures, and to confirm the migration and implementation prior to wholesale rollout and usage. (Live Trial).

4.2 General

For each release tests are to be engineered with particular regard to the following areas of good general practice:

a) to ensure that as much testing as possible is performed as early as possible in the lifecycle, to reduce defect correction costs and avoid unwelcome schedule disruption late in the lifecycle.
b) to arrange for as much testing as possible to be of an automated and re-runable nature, to reduce regression test costs, to speed test execution times, and to avoid unnecessary levels of human error.
c) to validate the end-to-end technical architecture employed.
d) to demonstrate that the system meets the specified and agreed levels of functionality and performance, and so is fit for service.
e) to comply with the General Testing Policy [1].

5. APPROACH

5.1 In essence the approach here is one of staged, systematic verification, with progressive integration of the wide ranging system components, culminating in overall validation of the fully configured system within the Live environment. It is a continuous process, spanning the whole development lifecycle, commencing with test analysis and the production of formal test material, and progressing through to test execution, result checking, and defect removal.

The products are moved through distinct, separately planned and executed stages of testing, each designed to progress the products to a higher level of assurance. Once a stage of testing has been completed for a particular product set, and the test activity reviewed and signed off, then that product set moves to 'Assured Status' and is deemed to be ready for use in the next stage and for progressively wider integration with their co-operating product sets. As the majority of the products required to make up the system are to be provided by 3rd party suppliers, then the emphasis in testing is skewed heavily toward 'black box' techniques. That is, the detailed inner workings of supplied products are not examined and put under test, but rather their gross behaviour is verified in the context of the services they are required to perform.

5.2 It is therefore particularly important that the Business Requirement is firmly understood at the outset. Here, the Business Requirement is taken as being encompassed by the Pathway Baseline Functional Specification [8], the Customer agreed Acceptance Criteria [2] and the SLAs used to formalise the various non-functional requirements that may exist. These must be maintained under strict Change Control, with the testers being impacted for all changes.

Similarly it is important that an 'Independent Testing Group' approach be adopted to maintain objectivity in developing the 'black box' tests necessary to successfully verify and integrate the mixed product set. That is, that products be treated alike, irrespective of whether they be supplied by a third party or developed in-house, with formal product handover into testing being established via formal product acceptance mechanisms.

5.3 Given the extremely condensed timescales available, it is also important that test planning commences at the earliest opportunity, and despite the separation of development and testing streams introduced by the independent testing approach outlined above, testers must nonetheless be involved throughout the development lifecycle, starting with high level 'logical' test planning, based on the business requirements, and progressively developing these into 'physical' test scripts, so that the test material is ready and approved in good time for use in test execution when the products are handed over into test.

This is consistent with the general recognition that testing follows a lifecycle closely interwoven with that of development. This is best embodied in the well established lifecycle 'V' diagrams, where development progresses down the left leg of the 'V' with analysis, design and construction, and then up the right leg of the 'V' with unit test, function test and system test. The horizontal relationships between left and right legs indicate the test analysis, planning and preparation required against each phase.

5.4 There are three principal categories of verification and validation required on this programme. The **'Functional Conformance'** of the system must be evaluated. That is, on a purely functional level, confirming that the services required of the system by the Customers, as defined in the functional requirements, are being met. The **'Architectural Performance'** of the system, both software and hardware must be evaluated. That is, the innovative underlying system architecture employed, with extensive infrastructure software and mixed hardware platforms, must be trialed under stress. The **'Business Integrity'** must be evaluated. That is, with the close relationship between Customer and Service Provider here, and the requirement for User Acceptance and Live Trial components within the Operational Trial phase of the Programme, the relatively late development of user processes and procedures, and the complex migration and implementation necessary, it is necessary to demonstrate integrity across the breadth of the whole business system.

These three categories - Functional Conformance, Architectural Performance, and Business Integrity - each have their own test lifecycle associated. That is not to say that they are conducted as separate activities, but rather that the test processes employed at each stage must take account of each aspect and their particular objectives and dependencies. In fact there is a clear overlap between the three when the lifecycles are mapped against the stages of testing, with Product Acceptance Test and System Test serving the objectives of both Functional Conformance and Architectural Performance, and with Integration Test similarly serving both Architectural Performance and Business Integrity.

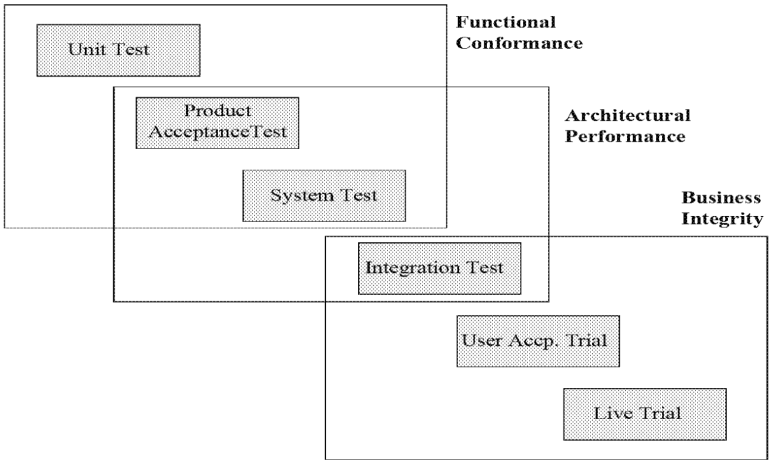


Figure 2 - Mapping of Testing Lifecycles onto the Stages of Testing

5.5 Functional Conformance.

In parallel with formulating the Pathway Baseline Functional Specification [8], which also serves as the Requirement Catalogue, test analysis should be conducted and initial

Business Threads produced describing very high level business driven test scenarios which will form the backbone of the System Test stage. This activity should be carried out in close co-operation with the Requirements Analysis activity, and with strong involvement from the Customer. Once the Pathway Baseline Functional Specification [8] is agreed and in place, more detailed test analysis can be afforded, and the System Test Strategy (STS) and High Level Test Plans (HLTPs) for System Testing can be drawn up by expanding the Business Threads. Again this should be done with strong involvement and sign off from the Customer.

Once the Product descriptions and Detailed System Specifications, consistent with the Pathway Baseline Functional Specification [8], are available, then further detailed test analysis of a much more specific nature can be conducted, engineering tests for use in Product Acceptance Test. These are intended to be used in verifying that each Product delivered into test can be shown to be compliant with their detailed specifications. The Product Acceptance Test Strategy is. produced and agreed.

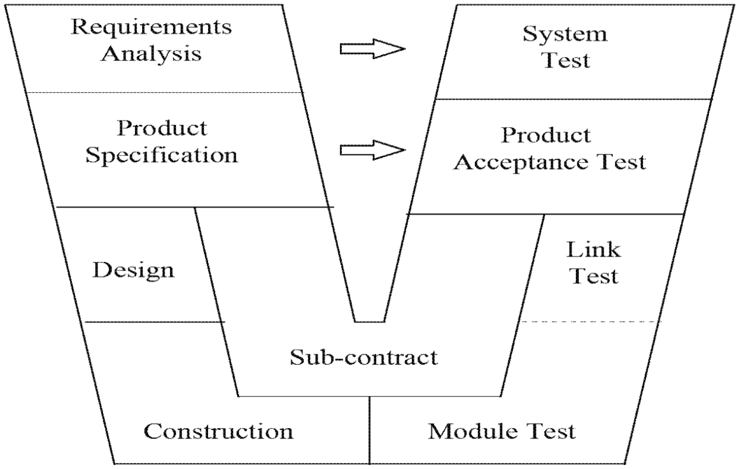


Figure 3 - The 'Functional Conformance' Testing Lifecycle

For those products that are developed in-house, the detailed design documentation should include detailed test plans for Unit Test, covering both individual module testing and the link testing of modules to form products. These should be consistent with the Unit Test Strategy produced for that product. Following construction the module and link test plans are carried out in Unit Test. For sub-contracted products, it is expected and assumed that the sub-contractors bring their products up to an equivalent state. From here on all products are regarded the same whether in-house or sub-contracted.

As each product is completed it is placed under Configuration Management control and delivered for independent testing where it is subjected to Product Acceptance Test according to the Strategy and Test Plans drawn up against the Product Descriptions and Detailed System Specifications. Any products not making the grade are rejected for correction and redelivery by the party concerned.

Once the requisite products have passed Product Acceptance Test they enter System Test, where, following the System Test Strategy and High Level Test Plans produced earlier, they are subjected to the Business Thread based tests to ensure conformance with the Requirements. Again any defects found are formally recorded, returned for correction as appropriate, and retested. System Test takes place on a simple hardware platform, with Interfaces out to external systems being simulated or truncated as appropriate. As much of the Infrastructure software will be included at this time as can be accommodated given the constraints of the platform. Some of the Infrastructure/Hardware interfaces will have to be simulated or truncated also.

The primary concern here is to achieve near end to end verification of the system and business flows, but early exposure of the infrastructure is desirable. Up to this point it is not envisaged that much if any Customer involvement would be required in the test execution.

5.6 Architectural Performance.

From the technical platform specification, which is derived during and immediately subsequent to establishing the overall requirement, and documented in the Pathway Baseline Functional Specification [8] and also in the Technical Environment Description [6] which includes coverage of the NFRs, security requirements, etc., the various technical attributes of the system can be considered and subjected to Test Analysis. Tests are engineered to verify these factors and HLTPs drawn up. Similarly, once the external system interfaces are formally defined, then (as part of System Test preparation, along with the Business Threads) the additional Inter-System Thread(s) can be derived, and their HLTPs produced accordingly.

Infrastructure software products required to satisfy the technical architecture are treated as described above for other products, again producing test plans from the Product Descriptions and the technical specifications. Again if any of these components are developed in-house, then Unit Test plans would be drawn up in the same way. The Unit Tests and Product Acceptance Tests would be executed accordingly.

When the System Tests were executed, albeit on a simple hardware platform, most if not all of the Infrastructure software would be present, and so the integration process of combining Application software and supporting infrastructure begins. All the APIs involved would be subject to verification, and the functional dependencies likewise. Simple aspects of application recovery and resilience would also be exposed here. Once the requisite System Tests are complete and the products concerned have reached Assured Status at this level, they are made available for wider use.

The Verification Centre (VC) collects the hardware, Infrastructure software, and Application software and subjects them to standard verification checks. The software is loaded, configuring the overall system according to the agreed System Configuration as documented in the CM Bill of Materials (CM-BOM) for that release. The system is thus packaged ready for for distribution. (These VC activities are limited to the 'counters' sub-systems.)

In parallel with these VC activities the system tested products are also taken by Integration Test (IT) where they are fully configured and properly installed on the full target hardware platforms in accordance with the CM-BOM.. From this point onward, an increasing level of Customer Involvement in test execution is required. Here, on a faithful hardware platform (full Live shape and significant size), the Integration Tests are run. These comprise not only the HLTPs engineered specifically to stress the architecture, but also the HLTPs corresponding to the Business Threads and the Inter-System Threads previously run in System Test. Here though the external interfaces are opened up to the BA/POCL systems and the Infrastructure software will now be running in a full operational environment and so can be tested end to end in co-operation with the supporting hardware.

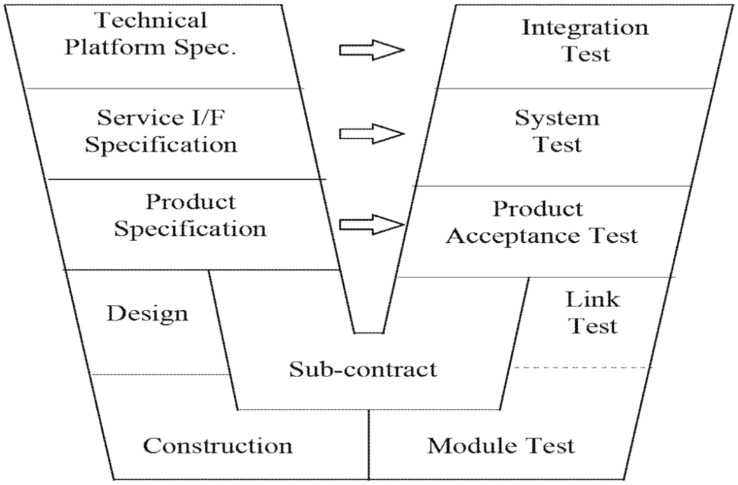


Figure 4 - The 'Architectural Performance' Testing Lifecycle

The primary concern here is to confirm that moving to the full platform has not caused the system to regress functionally, to confirm that the overall architecture hangs together end to end, to verify the recovery and resilience aspects of the system, and to subject the software and hardware to stress and load, measuring the performance achieved against the NFRs and SLAs. The principal source document here is the Technical Environment Description (TED) [6] which covers aspects such as the system and application architecture, performance, security, and other Non-Functional Requirements (NFRs). Any defects found will be subject to the same rigour as before.

5.7 Business Integrity.

This testing lifecycle is a little different. It is not interwoven with the development so much as with the production of the related business products which together with the IT system form the overall business system.

From the system requirements will come a full understanding of what work needs to be accomplished in the live field, and how that work will be conducted and fitted into existing work patterns. An interception plan can start to be drawn up as to how the first users of the system will take it on. What work will be included, over what period and with what support. (The period targeted here is known as the Live Trial).

Once the full set of support requirements have been determined, this plan can be expanded accordingly. This will cover such areas as the Help Desk, User Procedures and Instructions, the Training Programme and the support material required for it, and the User/Customer involvement in the actual migration to the new system, including manual fallback, etc.

In line with the Requirement Catalogue and consistent with the Pathway Baseline Functional Specification [8], Customer agreed Acceptance Criteria [2] will need to be drawn up and agreed. Likewise Service Level Agreements (SLAs) and related support contracts. From these documents plans should be drawn up and agreed as to exactly how the system will be trialed for acceptance - a User Acceptance Test (UAT.

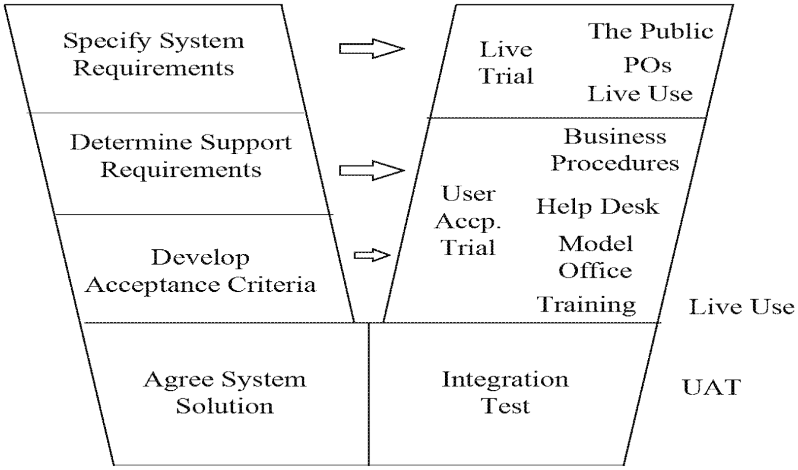


Figure 5 - The 'Business Integrity' Testing Lifecycle

When the overall system solution is agreed there are likely to be attributes of the technical specification of keen interest to the Customer from a business operations perspective, such as resilience, performance, recovery, etc. These will have been covered in the NFRs within the TED [6], and SLAs. When the Integration Test plans are reviewed and signed off, these attributes should be reviewed to confirm that the tests cover these issues adequately from a business perspective. The Customer, and perhaps key Users, should be involved in the test execution of Integration Test to be satisfied on these points. It is important that these areas are settled here and not left to the Mock and Live Trials to sort out, as any problems in these areas are likely to be expensive and time consuming to correct.

Once the Integration Tests are successfully completed, the users should have a very strong involvement in the User Acceptance Trial. This would be run in a 'laboratory' or 'model' office. This must be flexibly configurable so that at different times it can faithfully represent the various types of office that exist in the field, large and small. The tests engineered against the Acceptance Criteria and the SLAs are run exhaustively, with any remaining defects found being evaluated critically regarding their acceptability (and for how long) in the live. Fixes should only be applied at this stage where the defect is of a critical nature, as large numbers of fixes here would only serve to destabilise the system as a whole and jeopardise the implementation.

Finally, following system acceptance in the User Acceptance Trial, the system passes into the Live Trial - a period of genuine live operation but on a small scale and under careful control. Here the system would be installed in 20 to 30 selected and prepared representative Post Offices, for a period of 3 months.

5.8 This approach of progressively building up product sets, with products moving through distinct stages of testing, reaching 'Assured Status' and being employed in wider and wider integration of the growing system, results in a structured sequence of testing activities that can be planned and monitored by the Programme. Each stage centres on its own set of test objectives: they are focused. This reduces the likelihood of duplication of effort and allows greater efficiency in regression testing of changes, where dependent on the nature of the change, the correct point within the test lifecycle can be targeted.

Whilst in the initial development cycle these layers or stages of testing are likely to be carried out (it is desirable) by distinct teams of people, this is not necessarily the case later on in the maintenance phase where it is generally more cost effective to merge them. Similarly, the savings made in the development arena by using smaller more simulated environments in the earlier testing stages tend not to be so easily realised during maintenance, where it is generally simpler to do most of the testing in the larger more complete environments. stages, and possibly to operate more in complete environments. So typically the various tests required for a particular set of changes will be taken through by one team, one after the other, on a single platform. This also gives additional indirect savings, as CM overheads need only be incurred between different platforms.

The balance of cost versus risk is key to the management of the test planning and execution activities. As already indicated, regular checkpoints should be planned into the test suites to allow more focused retesting of fixes and regression testing following changes. Iterative running of tests can be made less troublesome, more cost effective and more consistent by utilising appropriate capture/replay tools (c.f. section 18 Test Environments). The production of expected results, and the subsequent checking of actual results against these is a prime area for consideration. Costs can quickly escalate here and become prohibitive. Pragmatism is the watchword. Being pragmatic about the creation of expected results can save a fortune, and provide better quality testing in the long run. A common pitfall is to strictly follow a process of painstakingly predicting the fine detail of the physical data output by a planned test. Doomed to fail for anything but the trivial process, this activity degenerates into one of testing ones ability to correctly project the data output, rather than putting the effort into testing the product concerned.

In the planning phase, expected results should for the most part be expressed in logical and not physical terms. Later, validated actual results can be used as physical expected results for subsequent regression test runs. Here again, use of standard utility sets to make the comparisons can reduce the effort involved and improve the accuracy of the checking.

5.9 There are strong schedule dependencies throughout test planning and execution on the development methodology, the release strategy and the programme schedule. It is important that testing and integration imperatives are taken into account in developing these schedules.

For example, infrastructure software (including middleware) and the data maintenance transactions for the key data elements will be required first, before any real progress can be achieved in assembling and integrating the products into working systems.

When the high level threads for System Test have been drawn up (c.f. section 11 - System Test) a schedule optimisation exercise will be carried out to establish these critical dependencies and confirm that the project schedules take account of them. These dependencies will be documented in the appropriate low level test strategies, and monitored throughout. Depending on the nature, number and complexity of these dependencies, it may also be decided to draw up a detailed product interception plan. The objective is to make it possible to successfully co-ordinate product delivery and testing, and to allow safe overlap of these lifecycle stages in accord with these critical dependencies, and so minimise the overall elapsed time for the programme.

6. RELEASE STRATEGY

6.1 The Release Strategy describes the phased release of system functionality into the Live arena. The Configuration Management Strategy [10] details how these releases will be controlled and their integrity (separate identity) maintained. It is important that each major release be carefully planned, and that versions of test material corresponding to each release be maintained in line with the versions of software and hardware products comprising that release, throughout the testing lifecycle.

6.2 To summarise the Release Strategy, it details three main phases:

- A Pre-release Pilot exercise, to be implemented at about 10 selected Post Offices in the Stroud area of England, offering restricted functionality in a carefully controlled and closely supported system regime. This is planned for September 1996.
- Release 1 - the main release, fully functional, robust and nationwide. This is planned for July 1997.
- Release 2 - a follow up release, content as yet unspecified. This is planned to follow 3 months after Release 1.

It should be noted that the special nature of the Pre-release Pilot means that acceptance of this release by the Customer must operate against a subset of the full Acceptance Criteria. Likewise, there is a high level of synergy between this and the full Live Trial. When the BRTS for the Pre-release Pilot is drwn up, it may be decided to omit the Live Trial stage to avoid duplication of effort and unnecessary delays. Finally, with only three months gap between Release 1 and Release 2, and given that the rollout process must inevitably consume a proportion of this time, considerable care will be needed in scheduling the testing and integration activities for Release 2. This will be reflected in the BRTS for Release 2, and provide details of any risk mitigation deemed to be necessary.

7. DEPENDENCIES

This section documents the principal areas of dependency upon which this strategic approach for testing and integration activities in Pathway relies. These will be maintained as required, and will be monitored throughout by the Testing and Integration Manager to ensure that they are satisfied in a timely manner and not allowed to adversely impact test progress.

(At a more detailed level, specific dependencies have been documented as appropriate against each stage of testing, under sections 9 through to 14.)

a) Testing will be organised as a separately managed group within Pathway for all testing from Product Acceptance Test onward - an Independent Testing Organisation.

b) The Risk Response Catalogue [9] contains various declared risk mitigations in terms of proving/testing activities. These will be included in the BRTS and cascaded down into the relevant Test Strategies dealing with that activity to ensure they do get addressed. A Risk review will be conducted to confirm that this has taken place..

c) Configuration Management practices and tooling will be implemented sufficient to support the testing activities in a phased release regime without adding too great a burden given the elapsed time pressures which apply.

d) The Configuration Management System will support the version control and environment build activities of the layered and iterative testing approach described.

e) Formal change management will apply for all key documents which act as input to test analysis. (For example, where detailed design for a sub-contracted product rightly and necessarily diverges from that of the Product description and/or Detailed System Specification, then that Product Description and/or Detailed System Specification will be subject to formal change management procedures and maintained in line with the product to be delivered - before it is delivered.)

f) All significant APIs (those between Products or between Pathway and external systems) will be rigorously documented at the outset.

g) Performance Budgets (Targets, derived from the TED and the relevant SLAs) will be set by the Technical Manager at the outset.

h) Sub-contractors will remain responsible for the in-programme and ongoing maintenance of products initially sub-contracted to them, and will actively participate in the testing activities described so as to minimise defect correction turn around times.

i) The CM-BOM will be maintained under strict CM control, and will be used as the basis of environment builds. As such either a separate BOM will be produced for each environment type from Integration Test onward, or the single document will cover each type as a distinct configuration.

j) Help Desk systems will employ the existing HCI styles pertaining to that area. BA staff will access the Pathway systems via CAPS and so employ that HCI style. POCL staff will access the Pathway systems via the counters interface for which an HCI style will be defined and agreed at the outset. This will result in only a single HCI being presented to any individual target user of the overall set of systems.

k) Activities at the Verification Centre will be conducted using the real application, having passed System Test and Integration Test, before the products are released further afield. This implicit dependency will be taken into account in the Programme Schedules.

l) Failure rates for products entering System Test will be defined and agreed at the outset, and monitored closely.

m) BA/POCL systems will include test facilities appropriate for linking with Pathway as described for Integration Testing and User Acceptance Trial stages. (Test to Test status). In the Live Trial links will be Live to Live.

n) External systems involved at each release (e.g. Gas) will in good time for testing of that release make suitable test facilities available for linking to the Pathway test facilities for the Integration Test and User Acceptance Trial stages. (Test to Test status). In the Live Trial links will be Live to Live.

8. RESPONSIBILITIES

8.1 This section lays out a general framework of responsibilities for testing of the Pathway system. The terms and roles used are intended to be generic rather than reference individual job titles which may well change during the course of a large project. For example, the roles of Quality Manager and Risk Manager may well be encompassed by the single job title Director of Quality and Risk Management. This framework is in two principal layers - the various general areas of responsibility that apply, and the more direct responsibilities that exist for specific testing activities.

8.2 General

Role	Responsibilities	Keywords
Customer	Review/Sign-off test plans for scope of tests. Be involved in test planning and running later stages of testing. Accept System.	Coverage Acceptance
Programme Manager	Review/Sign-off General Testing Policy. Allow for testing dependencies in project plans. Sign-off System for Customer Acceptance.	Policy Project Schedules System Acceptance
Technical Manager	Set performance targets. Review/Sign-off test plans for performance and technical integrity.	System Performance Technical Integrity
Operations Manager	Set operational requirements. Review/Sign-off test plans for service management and operability. Close involvement in running of later stages.	Handovers/Rollout Service Management Operability
Quality Manager	Apply QMS in testing area. Confirm handover process. Sign-off strategies. Check conduct of tests. Collect and interpret quality metrics. Check quality records. Sign-off test stages.	QMS & metrics Handovers Strategies & Conduct Quality Records
Risk Manager	Conduct risk assessments. Approve cost versus risk evaluations. Sign-off test plans. Produce residual risk report. Input to acceptance process.	Coverage Schedules Acceptance
CM Manager	Conduct handovers. Review environmental status against CM-Bomb. Confirm integrity of configurations.	Handovers CM-Bomb Environments
Security Manager	Review / Sign-off test plans for security and access coverage. Confirm retention periods for key documents / products. Check security status for different environments / data usage.	Access Retention Environment / Data
Suppliers and Workforce	Ensure acceptable product quality on supply. Support testing process throughout life-cycle and provide adequate maintenance turnaround.	Product Quality Support
Testing & Integration Manager	Sign-off Strategies and plans to ensure adequate test coverage in all respects. Project Management and general co-ordination of testing and integration activities to schedule and budget, monitoring dependencies. Implicit responsibility for interfacing with all above areas.	Coverage / Conduct Strategies / Plans Testing Schedules Environments / CM Quality / Security Acceptance

8.3 Specific

Refer to the detailed process descriptions in the relevant section of the Testing Processes document [11] for specific responsibilities of testers / team leaders / etc. for each product activity relating to a particular stage of testing.

9. UNIT TEST

9.1 Context

Performed for in-house development products only. Performed by Pathway with no Customer Involvement. Regarded as part of development. Applies to individual modules and their interlinking within a product. Formally planned as part of design, but not necessarily scripted in detail. Follows the Unit Test Strategy produced for that product. Informally executed, using debug facilities etc. as appropriate. Results formally recorded and retained. First part of Functional Compliance Test Lifecycle. Two phases - Module Test and Link Test (c.f. Unit Test process in Appendix 1).

9.2 Objectives

To demonstrate that each module developed in-house conforms with the detailed module specification.

To pre-integrate co-operating modules within a product by demonstrating that their APIs are implemented correctly and so that the modules link together as specified and co-operate properly as an integrated unit.

To make the components of a product ready for Product Acceptance Test such that in 90% or more of cases, the product will not be rejected and require reworking.

To demonstrate that the unit conforms to the relevant HCI where applicable.

9.3 Overview

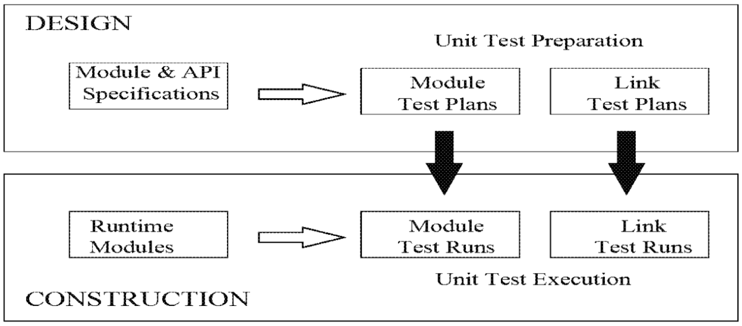


Figure 6 - Schematic Overview of Unit Test

9.4 Inputs/Outputs

Inputs: Module Specification and API specifications.

Outputs: Diary of Testing, Module Test plans, Link Test Plans, Test Results, Defect/Resolution Log, Pre-integrated functionally conformant modules.

9.5 Dependencies

Formal Code Walkthroughs are performed prior to Unit Test, excepting where Rapid Application Developemnt (RAD) approach has been adopted as the development methodology.

Test tools and environments as identified in the Unit Test Strategy are made available in good time. (For Unit Test these may be no more than standard workbench facilities like interactive debugging runtime tools.)

10. PRODUCT ACCEPTANCE TEST

10.1 Context

Performed for all bespoke software products, irrespective of whether developed in-house or subcontracted. (In accordance with the General Testing Policy [1] this does not apply to standard proprietary items of hardware and software, such as MS Windows NT, or a laser printer, where it is expected that the product will perform according to the manufacturer's specification.) Performed by Pathway with no Customer involvement. Regarded as independent of development. Formally planned and scripted. Follows Product Acceptance Test Strategy. Formally executed, preferably in automated, re-runable fashion. Test results formally recorded and retained. Follows Unit Tests for product concerned. Forms middle stage of Functional Conformance Test Lifecycle and first stage of Architectural Performance Test lifecycle. Precedes the Operational Trial phase.

10.2 Objectives

To demonstrate that each product delivered into test conforms to the Product Descriptions and/or Detailed System Specifications for that product, and so is fit for wider testing use.

To confirm that the Human Computer Interface (HCI) employed conforms to that designated as appropriate that that product set.

Forms formal acceptance of products from sub-contractors and in-house developers alike.

0.3 Overview

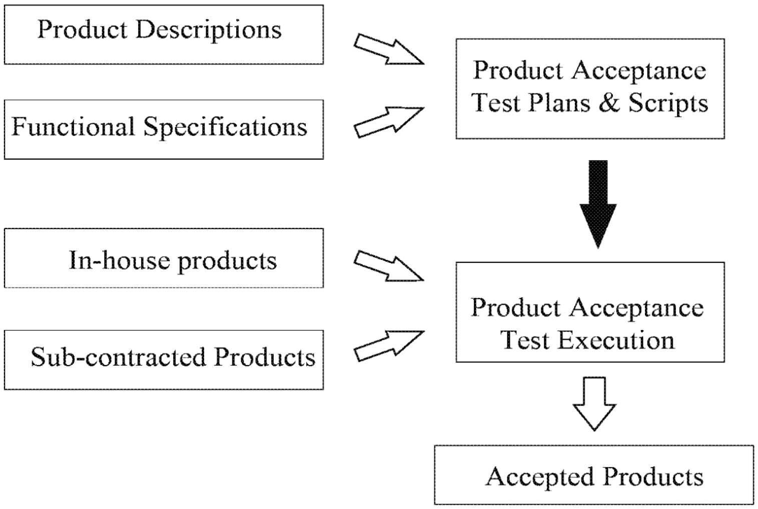

Product Descriptions
Functional Specifications
Product Acceptance Test Plans & Scripts
In-house products
Sub-contracted Products
Product Acceptance Test Execution
Accepted Products

Figure 7 - Schematic Overview of Product Acceptance Test

10.4 Inputs/Outputs

Inputs: Product Descriptions, Detailed System Specifications, Unit Tested Code

Outputs: Product Diary of Testing Activities, Acceptance Test plans and scripts, Test Results, Product Rejection/Correction Log, Defect/Resolution Log, Functionally conformant Products.

10.5 Dependencies

Products will be delivered into this stage in a timely manner and will be of high quality, and on the whole be functionally conformant with their specifications. There is an expectation that no more than 10% of products will need to be rejected here.

11. SYSTEM TEST

11.1 Context

Performed on receipt of accepted products from Product Acceptance Test area. Performed by Pathway with little or no Customer Involvement. Regarded as separate from development. Formally planned and scripted. (Scripts reused in Integration Test stage.) Follows System Test Strategy for that release, and is based on Business Threads and Inter-System Threads formulated with and agreed by the Customer. Formally executed, preferably in re-runable fashion. Run on simple single platform environment. Test Results formally recorded and retained. Final stage of Functional Conformance Test Lifecycle, and second stage part of Architectural Performance Test Lifecycle. Forms the first stage of the Operational Trial phase.

11.2 Objectives

To demonstrate through a series of comprehensive business driven scenarios that the software system as a whole functionally conforms with the agreed Requirements.

To expose the Infrastructure software to use by the application, and to demonstrate that within the limitations of a reduced hardware platform it provides the application with the specified support.

To perform initial verification of OHE requirements where available equipment, stationery and other materials allow.

To confirm that the HCI employed by each product, when run in conjunction with each other, does not result in a clash of HCI styles.

To expose the software system, and hardware as available and appropriate, to both internal and external representative bodies for minority groups and dis-advantaged peoples (e.g. Help the Aged). Specifically to seek positive feedback on any potential problems in this area at the earliest possible stage.

To demonstrate simple recovery and resilience features of the system within the bounds of the platform in use.

To form a comprehensive system regression pack for later use, and to prepare the way for full system integration in the following stages of testing.

11.3 Overview

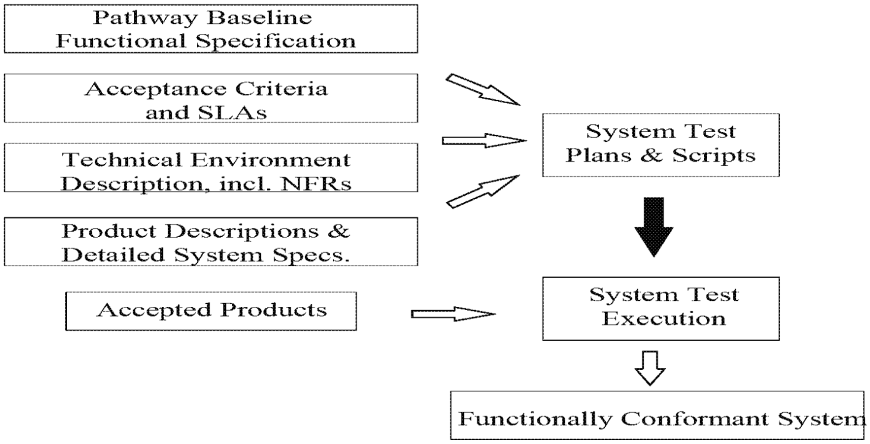

Pathway Baseline Functional Specification
Acceptance Criteria and SLAs
Technical Environment Description, incl. NFRs
Product Descriptions & Detailed System Specs.
Accepted Products
System Test Plans & Scripts
System Test Execution
Functionally Conformant System

Figure 8 - Schematic Overview of System Test

11.4 Runtime Considerations

Regular checkpoints will be planned in. These will where practical exploit natural quiescent points. Otherwise they will need to be engineered in. At the checkpoints all data should be preserved in a restorable form to allow the test suite to be restarted from that point at a later time should the need arise to retest that particular portion following a change to the baseline. This will help to keep the fix retest and regression test costs down to a bearable level and avoid wasting valuable elapsed time in unnecessarily re-running entire test suites.

Test execution should be ‘captured’ (c.f. recommendations for tools in section 18 Test Environments) for later automatic ‘replay’ when iteratively testing fixes or when regression testing following changes.

Where practicable standard utilities swill be used for checking actual results against expected results, rather than manually checking. This should reduce costs and improve accuracy. However, the approach to drawing up expected results should be pragmatic. In the first instance, except for the most simple of cases, they should be couched in logical rather than physical terms. First test runs can be checked against these manually. Once consistent for the most part, these actual results can be taken, edited if required, and serve as the physical expected results for subsequent iterations.

11.5 Inputs/Outputs

Inputs: Pathway Baseline Functional Specification [8], Acceptance Criteria [2], SLAs, NFRs from TED [6], Product Descriptions, Detailed System Specifications.

Outputs: Business Threads, Inter-system Thread(s), System Test plans and scripts, Defect/Resolution Log, Test Results, Functionally conformant system ready for use in Integration Test.

11.6 Dependencies

The Tools and environments indicated in the System Test Strategy are available for use in good time.

Functionally conformant products delivered in the appropriate sequence to service the running of the threads in the optimum fashion.

12. INTEGRATION TEST

12.1 Context

Performed against a fully configured and properly installed system as specified by the CM-BOM, following completion of System Test. Performed by Pathway with moderate involvement from Customer. Regarded as separate from development. Formally planned and scripted, including significant reuse of System Test material. Follows Integration Test Strategy for that release, and based on HLTPs developed from SLAs and NFRs, together with Business Threads and Inter-System Threads carried forward from System Test. Formally executed. Simulated loads re-runable. Run on a faithful platform (full live shape and significant size) including all hardware and software variants, comms., peripheral equipment, etc. Interfaces to BA/POCL Test systems are opened up (with appropriately selected / fabricated data) to exercise full end to end data flow. Similarly, any external systems (e.g. Gas, Electric, etc.) appropriate to the release concerned will be connected. (Assumed the agency concerned will have appropriate test / simulation facilities.) Test Results formally recorded and retained. Forms final stage of Architectural Performance Test Lifecycle and first stage of Business Integrity Test Lifecycle. Second stage of the Operational Trial phase.

12.2 Objectives

To integrate total software and hardware system, verifying the system architecture on an end to end basis and verifying the integrity of the CM-BOM which will be used later as the blueprint for building and rolling out the system into Live.

To exercise end to end data flow, including BA/POCL and external agency systems as appropriate, using selected / fabricated test data.

To confirm that transit to the full target environment has not caused the system to regress functionally.

To expose the full Infrastructure software to use by the full application, and to demonstrate that in conjunction with the full hardware platform they provide the application with the specified support.

To perform full verification of OHE requirements with all necessary equipment, stationery and other materials, including volume, performance and accuracy aspects of its operation.

To confirm that the HCI employed by each new product introduced (e.g the BA/POCL systems, etc.), when run in conjunction with the Pathway products, does not result in a clash of HCI styles for a given target user.

To expose the software and hardware system, to both internal and external representative bodies for minority groups and dis-advantaged peoples (e.g. Help the Aged). Specifically to seek confirmation that earlier feedback has been taken into account and to take further feedback on any potential problems remaining in this area.

To commence verification of the operability aspects of the system (completed in the User Acceptance Trial stage).

To confirm the performance attributes of the total system, verifying that the SLAs and NFRs (as defined in the TED [6]) have been met.

To ensure that all standard proprietary products (which are assumed to operate in accordance with the manufacturers specification) are used fully within the context of the overall system, and provide the functionality and support services expected of them.

To confirm that the specified security aspects of the system (from the TED [6]) operate with integrity on the target live hardware platform.

12.3 Overview

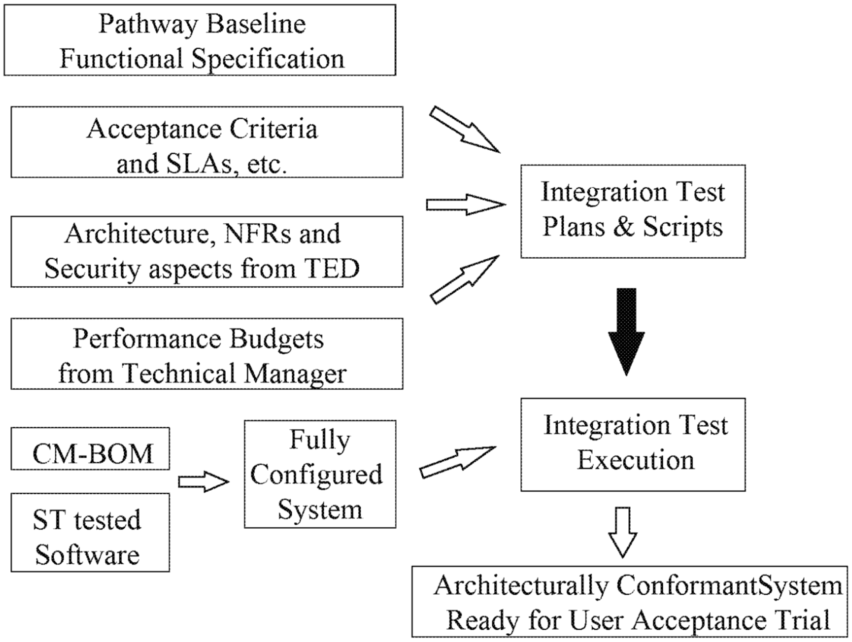

Pathway Baseline Functional Specification
Acceptance Criteria and SLAs, etc.
Architecture, NFRs and Security aspects from TED
Performance Budgets from Technical Manager
CM-BOM
ST tested Software
Fully Configured System
Integration Test Plans & Scripts
Integration Test Execution
Architecturally ConformantSystem Ready for User Acceptance Trial

Figure 10 - Schematic Overview of Integration Test stage

12.4 Runtime Considerations

Regular checkpoints should be planned in. These should where practical exploit natural quiescent points. Otherwise they will need to be engineered in. At the checkpoints all data should be preserved in a restorable form to allow the test suite to be restarted from that point at a later time should the need arise to retest that particular portion following a change to the baseline. This will help to keep the fix retest and regression test costs down to a bearable level and avoid wasting valuable elapsed time in unnecessarily re-running entire test suites.

Test execution should be 'captured' (c.f. recommendations for tools in section 18 Test Environments) for later automatic 'replay' when iteratively testing fixes or when regression testing following changes.

Where practical standard utilities should be used for checking actual results against expected results, rather than manually checking. This should reduce costs and improve accuracy. However, the approach to drawing up expected results should be pragmatic. In the first instance, except for the most simple of cases, they should be couched in logical rather than physical terms. First test runs can be checked against these manually. Once consistent for the most part, these actual results can be taken, edited if required, and serve as the physical expected results for subsequent iterations.

12.5 Inputs/Outputs

Inputs: Pathway Baseline Functional Specification [8], Acceptance Criteria [2], SLAs, the TED [6] (detailing the system architecture, NFRs, performance and security aspects), Performance Budgets derived from the SLAs by the Technical Manager, System Tested fully configured and properly installed system with full target live hardware platform.

Outputs: Integration Test plans and scripts (in addition to the existing System Test plans and scripts), Defect/Resolution Log, Fully integrated and operationally functional system ready for use by the Verification Centre and for entry to the User Acceptance Trial.

12.6 Dependencies

The tools and environment as indicated in the Integration Test Strategy will be available for use in good time. (These are likely to include system performance monitoring facilities and simulated load generation engines for each of the hardware types making up the overall configuration.)

The necessary BA/POCL test systems exist, are available for use, and are compatible.

The necessary external agency test systems exist, are available for use, and are compatible.

13. VERIFICATION CENTRE

13.1 The Verification Centre is an established standard facility within ICL for preparing and pre-validating configurations, packaging them ready for distribution. Pathway will exploit this valuable facility for assembling the various hardware components, configuring them correctly, loading the various software components, configuring them properly, and running simple checks to confirm that all is operating in accordance with specification. It operates in conjunction with Configuration Management to establish Software and Hardware Baselines, or Builds. The Configuration Management Bill of Materials (CM-BOM) in exploded form acts as the specification for this task.

13.2 The Verification Centre fits in between the Integration Testing and User Acceptance Trial stages, and builds the configurations for delivery to all later stages in the lifecycle, including Live. Verification will be performed using the actual applications, as passed by System Test and Integration Test, before distribution to the various target environments. (Schedules must allow for this.)

The software products once passed by Product Acceptance Test are taken, under CM control.They are made available not just to System Test (ST) but also the Verification Centre for early use. All fixes follow the same route throughout ST and Integration Test (IT), so at the end of IT the Verification Centre will be operating against the same code-set that passed by IT, and a final pass of the verification can be made against this to establish the first formal baseline (or build) before passing it, under CM control, to the User Acceptance Trial (UAT) in the Model Office. The same principles then apply for all the later stages through to Live operation. The following schematic illustrates. (Note - the VC activity is limited to the 'counters' sub-systems.)

13.3 Overview

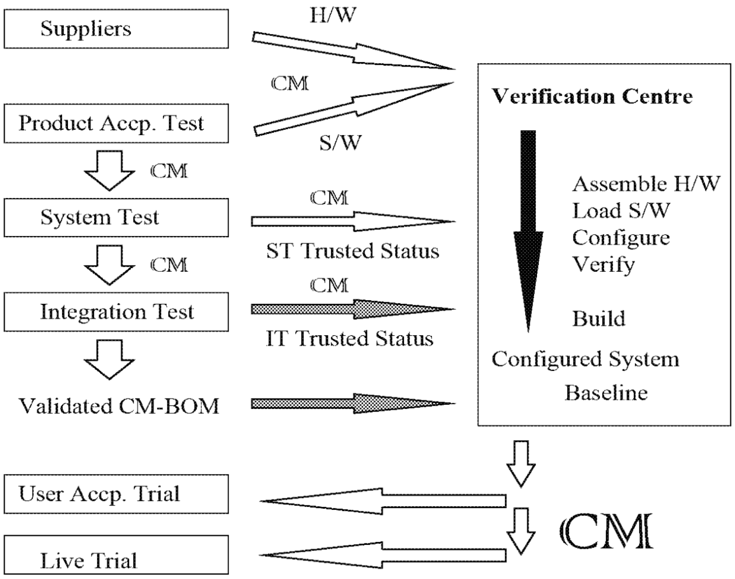


Figure 9 - Schematic Overview of Verification Centre Role

13.4 Should any product delivered into the VC be found to be faulty, or if incorrect CM control renders it so, then it will be rejected, and a corrected version awaited from the supplier concerned. Similarly, should the configured system delivered from the VC into the User Acceptance Trial be found to be faulty with respect to the hardware or the configuration, then again it will be rejected for the VC to correct. (It is important that the failure rates here be kept very low as the disruption caused would otherwise be great, particularly on schedules.)

13.5 Dependencies

The CM-BOM will be a complete configuration map covering all soft variables, such as Operating System defaults, INI file settings, etc.

The VC will confirm all such configuration settings are as specified.

Project Schedules will reflect the need for the VC to use the actual application in its later stages of verification, prior to release to Integration Test.

Failure rates will be held below the following thresholds:

into the VC	no more than 1% for hardware and proprietary software
	no more than 0.5% CM handover errors
from the VC	no more than 0.1% during rollout

14. USER ACCEPTANCE TRIAL

14.1 Context

Performed on receipt of fully configured and properly installed system from the Verification Centre, following completion of Integration Test. Performed jointly by Pathway and the Customer. Formally planned and partially scripted, being driven by the Customer agreed Acceptance Criteria [2] as agreed previously. Executed primarily by Customer staff, following Business Procedures/Instructions, in a ‘Model Office’ environment, running on a fully operational platform faithfully representing the Live situation, with little or no simulation. This will include interfaces open to BA/POCL and external agency test systems. These should be fully configured systems and if security considerations permit they should employ Live data sets. Forms the 2nd stage of the Business Integrity Testing Lifecycle and the penultimate stage of the Operational Trial phase.

14.2 Objectives

To demonstrate the operational viability of the overall business system.

To serve as the vehicle for the formal acceptance of the system by the Customer, driven by the Customer agreed Acceptance Criteria [2] - in effect a ‘User Acceptance Trial’.

To exercise the Help Desk operation and other support routes.

To serve as the final validation of OHE operation, and to prove the final versions of Live stationery and other materials prior to bulk order.

To integrate the Business Procedures/Instructions with the IT System, and to verify their joint operation.

To identify any major shortcomings in this joint operation and to overcome them either by correction or interim management, as appropriate.

To validate the end to end rollout process and Configuration Management mechanism, for each principal office type, in respect of both hardware and software, including a ‘Dress Rehearsal’ of the Implementation Plan for the first release.

To confirm correct operation of VC and CM delivery and distribution services across the range of system configuration variants that apply (e.g. the various post office types, different versions of software and hardware, different builds and releases, etc.).

To confirm that the HCIs (against which the system components have already been validated) deliver the appropriate levels of usability.

14.3 Overview

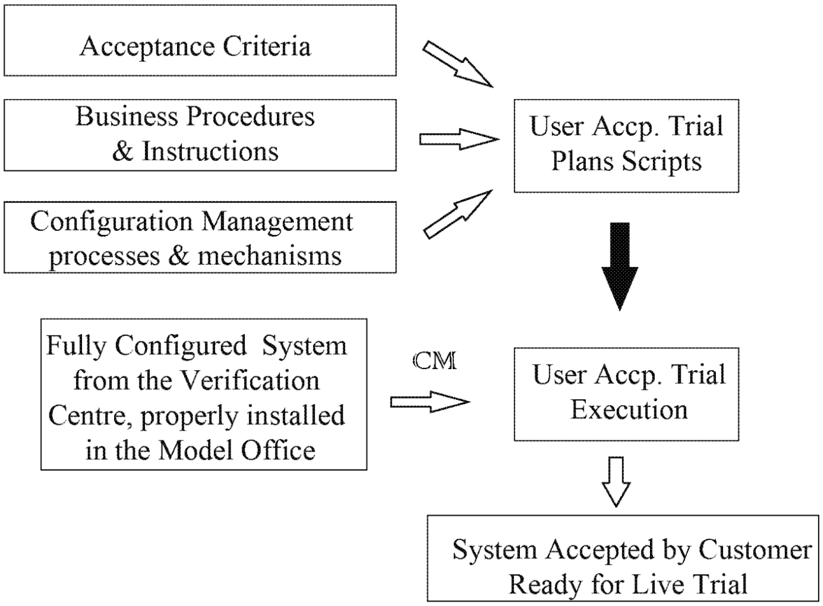

Acceptance Criteria
Business Procedures & Instructions
Configuration Management processes & mechanisms
User Accp. Trial Plans Scripts
Fully Configured System from the Verification Centre, properly installed in the Model Office
CM
User Accp. Trial Execution
System Accepted by Customer Ready for Live Trial

Figure 11 - Schematic Overview of User Acceptance Trial stage

14.4 Inputs/Outputs

Inputs: Customer agreed Acceptance Criteria [2], Business Procedures / Instructions, CM processes and mechanisms, fully validated IT System (passed Integration Test), fully configured system installed in the Model Office.

Outputs: User Acceptance Trial plans, supported by scripts where appropriate, Defect/Resolution Log, System Accepted by Customer ready for Live Trial, Validated roll-out and CM process.

14.5 Dependencies

Business Procedures / Instructions ready for use in good time for the planning stage.

Model Office setup agreed in good time for the planning stage, and prepared in good time for the execution stage.

Appropriately skilled and trained Customer staff available for the execution stage.

Full OHE, stationery and other materials in place and verified in good time for final validation through the User Acceptance Trial.

Help Desk configured and appropriately staffed and trained for initial exposure in User Acceptance Trial prior to Live running.

Rollout and CM processes and mechanisms defined, in place and verified in good time for final validation through the User Acceptance Trial.

15. LIVE TRIAL

15.1 Context

Performed following formal acceptance of the system by the Customer in the User Acceptance Trial. Performed by the Customer with strong Pathway support. In effect a Live Pilot. The full operational business system run Live in 20-30 Post Offices for a period of 3 months, with full support arrangements in place. Close control exercised throughout.

15.2 Objectives

To serve as a Live Pilot before wider use across the UK.

To integrate all supplementary material and activities with the IT system as a total business system.

To validate the training material and processes.

To validate the support arrangements, including Help Desk operation.

To validate the Migration and Implementation activities in a limited and closely controlled sphere prior to wider application across the UK.

To establish initial reaction to the system from Post Office staff and the general public, and so to provide an early warning of potential difficulties in a limited sphere of exposure.

15.3 Overview

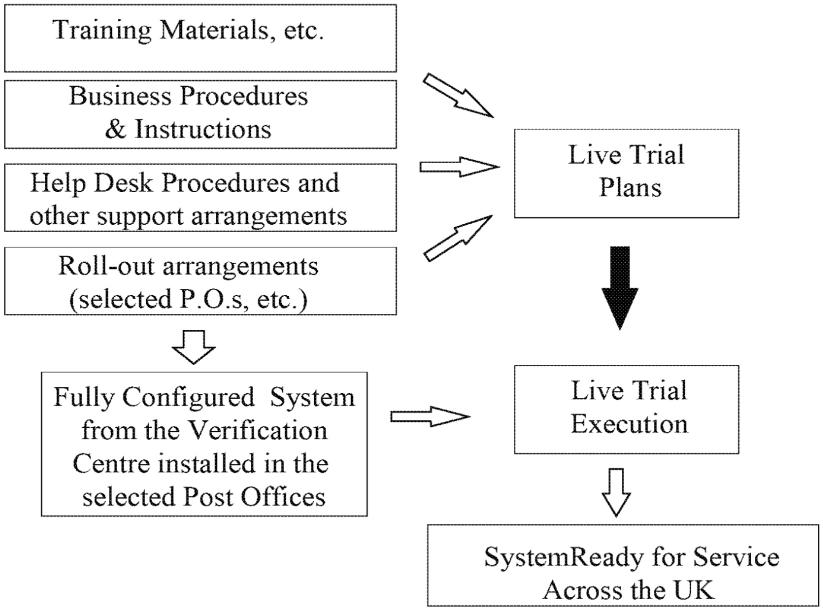

Training Materials, etc.
Business Procedures & Instructions
Help Desk Procedures and other support arrangements
Roll-out arrangements (selected P.O.s, etc.)
Live Trial Plans
Fully Configured System from the Verification Centre installed in the selected Post Offices
Live Trial Execution
SystemReady for Service Across the UK

Figure 12 - Schematic Overview of Live Trial stage

15.4 Dependencies

Rollout arrangements agreed in good time for the planning stage, and be completed in good time for the execution stage. (To include selected Post Offices, their attributes, the staff involved, etc.)

Help Desk and other support arrangements agreed in good time for the planning stage, and to be in place in good time for the execution stage. (To include any user/ops guides, instructions and procedures that are required, and staffing plans).

Training arrangements agreed, and supporting training material in draft in good time for the planning stage. Training material to be produced and training given for selected offices/staff in good time for the execution stage.

16. MAINTENANCE

16.1 Context

Performed in two phases of operation: re-test of errors found in later stages of testing; re-test of errors/enhancements after system has gone live. Regression packs identified on completion of each phase of testing, consisting of a subset of the overall test scripts for that phase. Errors found in later phases of testing will be re-tested at entry level (Unit Test for in-house products, Product Acceptance Test for Third Party products), with re-testing at later stages also possible. During active testing, test packs will be rationalised at the end of each test stage. On completion of Integration testing for a release, scripts across phases will be rationalised into a release level regression test pack, with suitable tests selected from each test stage. Resolution of errors found subsequently in User Acceptance Trial activities or beyond will be assessed by testers, and selected tests amended/run to verify error has been resolved. Other selected tests will be rerun to ensure that the system has not been adversely impacted (regressed) by the changes.

16.2 Objectives

To provide an effective subset of tests from all phases of testing to allow focused validation of changes to the system (either error fixes or enhancements).

To reduce script maintenance costs by retaining tests which cover the full width of the system's functionality but without duplication.

To facilitate a single team managing maintenance testing of the system after it has gone live.

16.3 Overview

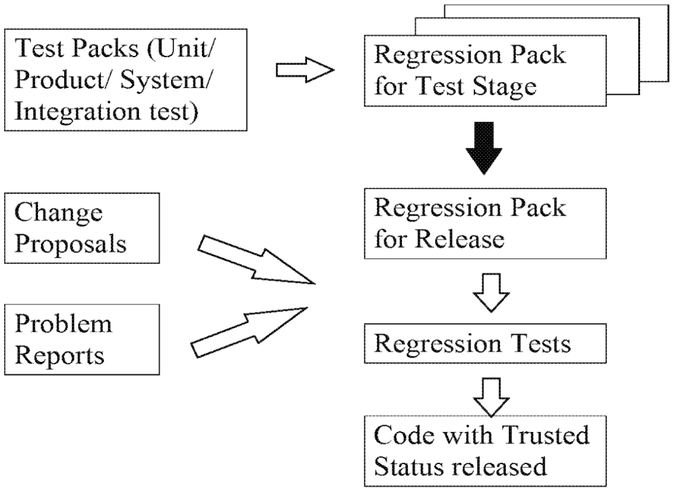

Test Packs (Unit/ Product/ System/ Integration test)
Regression Pack for Test Stage
Change Proposals
Problem Reports
Regression Pack for Release
Regression Tests
Code with Trusted Status released

Figure 13 - Schematic Overview of Maintenance Stage

16.4 Inputs/Outputs

Inputs: Test packs from Unit Testing, Product Acceptance Testing, System Testing and Integration Testing; Test Execution Plans from System/Integration testing; input data files; actual results; Change Proposals; Problem Reports

Outputs: Regression Test Pack for each Test Stage, single Regression Test Pack for a Release; Enhanced Test Execution Plans; Enhanced software validated effectively at minimum cost.

16.5 Dependencies

The workload for Independent Testing will contract to require only a small maintenance group.

17. EXTERNAL CERTIFICATION

Certain products employed by Pathway will require external certification before they can be used operationally, for legal reasons. For example, the weigh scales in conjunction with the PCs on the counter top will need to be examined and certified by the weights and measures authority before they can be legally used in live operation out in the Post Offices.

All such products will be formally identified for each release, together with the authorising bodies concerned.These will then be detailed in the BRTS for that release, and the External Certification activity will be included in the project plans, and monitored accordingly.

There is no direct technical dependency on this certification, but it has an obvious bearing on System Acceptance, and on the Live Trial.

18. TEST ENVIRONMENTS

18.1 General

This section covers the key attributes of the test environments as they apply to this general approach to testing. First the environments themselves are described, and then the supporting toolset. Whilst it is not essential to have sophisticated environments and tools to perform high quality testing, a certain level of automation is required. Specific environment and tool requirements will be detailed in the lower level strategies for each test stage. Here follows brief generic descriptions to set the context.

18.2 Environments

As testing progresses through the stages, from Unit Test (UT), through Product Acceptance Test (PAT), System Test (ST), Integration Test (IT), and on to the User Acceptance Trial (UAT) and Live Trial (LT), then so does the optimum environment type. In general, the environment will start off supporting a more private and isolated mode of testing - usually a single individual, with high levels of simulation and relaxed configuration control, and move through to a fully shared environment with little or no simulation and rigourous configuration controls.

UT environment - small, little data, harness based allowing simulated data and user input and preferably simulated calls to better isolate the modules concerned. Private file set and database, private online monitor (usually simulator to avoid high machine resource costs). Interactive mode of operation highly desirable (e.g. debug facilities). Little or no CM control. Not hardware specific. Many instances required, so optimise for low machine resource usage.

PAT environment - small, little data but enough to link related products, may be harness based but less important (simulated calls useful to overcome schedule dependencies). Private file set and database, private online monitor (simulator still preferable). On completion enters CM domain. Many instances required, so still optimise for low machine resource usage.

ST environment - medium size, medium data volumes, sufficient to link entire business threads, preferably no harness. No simulated calls, except in that all software and data sets are non-Live (test systems used throughout). File sets and database used in both private and shared modes. Real online monitor, not simulator. All software under CM control. Operates on minimal hardware set, condensed configuration, no architectural layering. Small numbers so do not need to optimise for machine resource usage.

IT environment - large size, large data volumes, sufficient to stress live configuration. No harness, except a performance engine to generate simulated online load. No simulated calls, except that test systems used throughout (no Live links). File sets, database and online monitor all shared. All software and hardware under CM control, configured to CM-Bomb. Full live hardware set, full configuration, full architectural layering. Single Instance. Able to act as each office type (configurable).

UAT environment - same as IT environment, but built by the Verification Centre (VC) in model (laboratory) office setting.

LT environment - Live, but with limited roll-out.

18.3 Tools

Given the timeframes involved for this programme, it is not sensible to consider deploying new generic test management tools or test utilities as. their usage takes some considerable time to mature and the learning curves are prohibitive. Similarly static fault finding tools tend to be platform and language specific (because they examine the source code) and so cannot generally be employed across a mixed platform set. Dynamic fault finding tools are likely to give the best return.

Of the numerous dynamic fault finding tools on the market, a shortlist will be drwn up which satisfy Pathway's principal platform types, and provide:

- Test drivers with capture / replay facilities for regression work
- Comparators for checking test results (including database)
- Interactive simulators (debuggers)

Each product will then be closely evaluated and demos run. The final contenders will be trialed in situ prior to purchase.

In addition to these generic tools, a test execution harness will be required for the Unit Test stage, and to a lesser extent for the Product Acceptance Test stage. This will be specified in the Unit Test Strategy and written in-house, for each principal platform type concerned. Where simple wraps will assist in the execution phase of other test stages, these will likewise be soecified in the detailed startegies and again written in-house.

APPENDICES

A1.TEST PROCESSES

A2.GLOSSARY OF TERMS

APPENDIX 1

TEST PROCESSES OVERVIEW

A1.1 Appendix 1 to the Testing and Integration Strategy provides an overview of the testing processes supporting the Pathway solution. Figure A1.1 below provides a high level overview of the main processes, their key inputs and inter-dependencies. The remainder of this Appendix provides an abridged view of the discrete processes which support each major test activity.

For a more detailed description or for details of the remaining processes see the Testing Processes document [11].

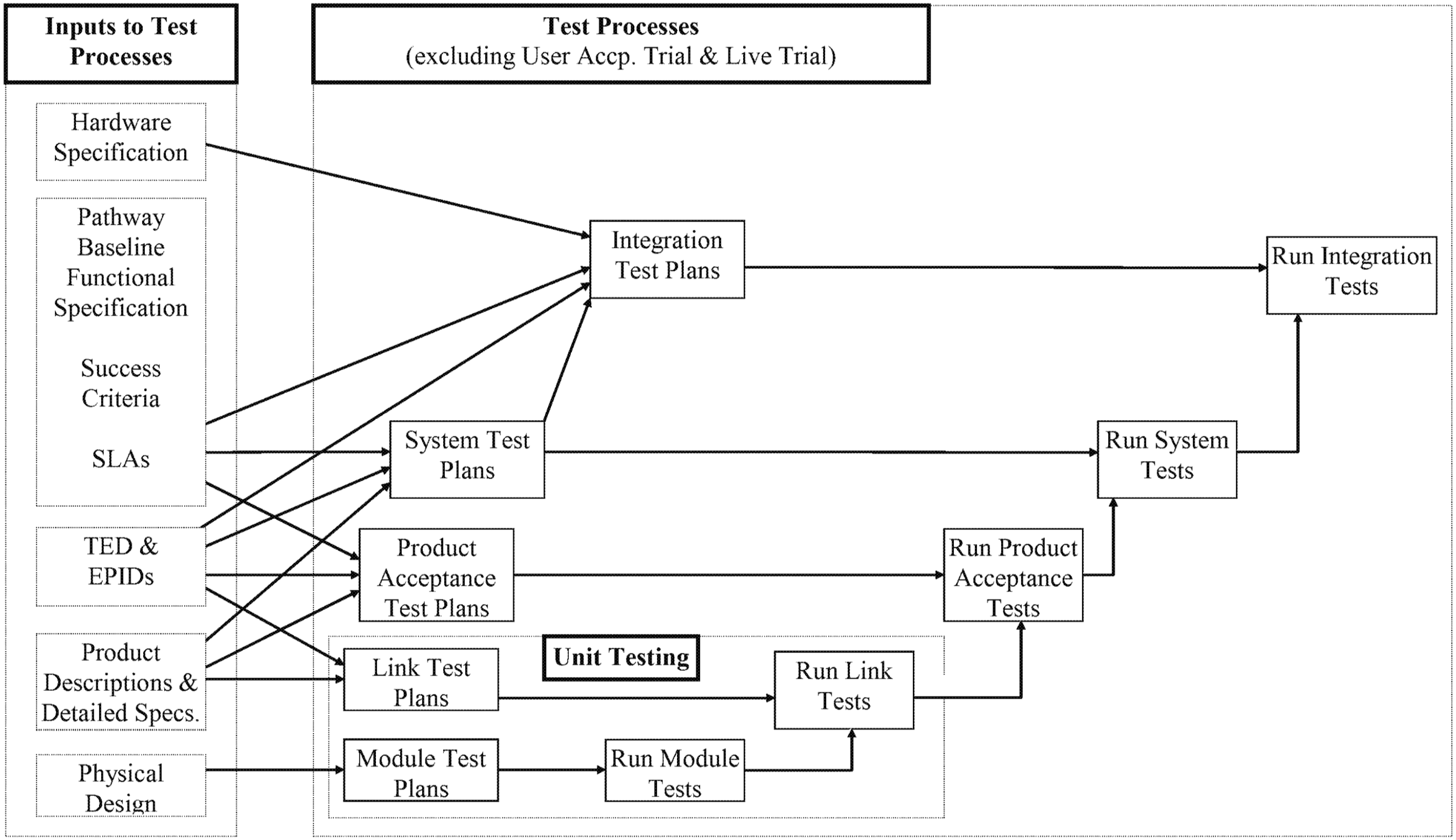


Figure A1.1 - Schematic Overview of Testing Processes

A1.2 UNIT TEST PROCESS

A1.2.1 Purpose

To 'white box' test a single product developed in-house. Unit testing includes validation of the discrete software modules (module testing) and their integration into a complete product (link testing) prior to delivery into Product Acceptance Testing.

A1.2.2 Activities

Produce Unit test strategy for this product;

For each software module, analyse the Physical Design and produce Module Test plans. While this testing is largely informal, the test analyst will need to ensure that there is a clear definition of the input data states, run instructions and expected results;

Working from the Physical Design, the Pathway Baseline Functional Specification [8], the relevant Product Description and relevant EPIDs, produce Link Test plans which fully validate the complete integrated product. Link Test plans should document input data states, run instructions, and expected results. Tests should also include cross references back to the design products, allowing auditability and traceability;

Run Module Tests, with tools and environment as per Unit Test Strategy;

Check Module Test results - initially that they are produced and retained but then subsequently checking against the previously retained set, using them as expected results;

Sign off Module Tests;

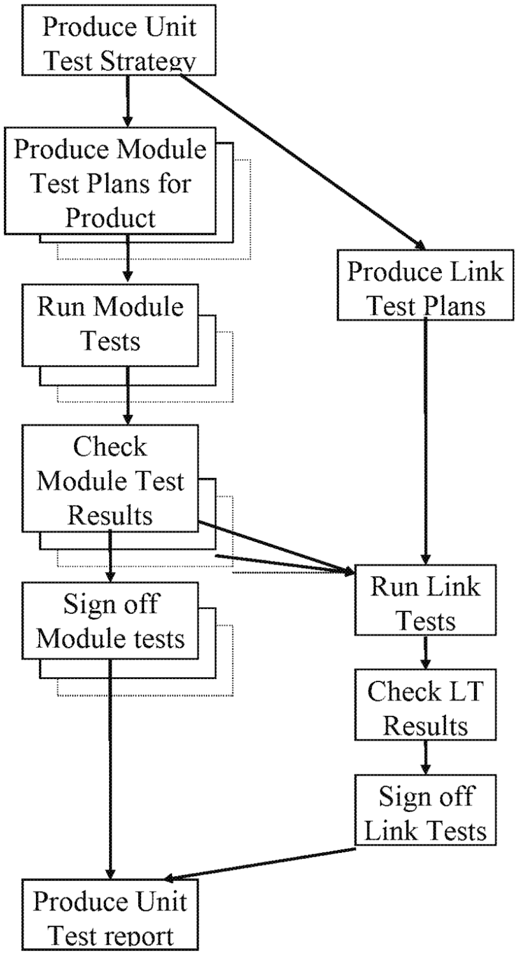
Run Link Tests, with tools and environment as per Unit Test Strategy;

Check Link Test results against expected results;

Sign off Link Tests;

Produce Unit Test Report for this product.

A1.2.3 Unit Test Activities Dependency Diagram



A1.3 PRODUCT ACCEPTANCE TEST PROCESS

A1.3.1 Purpose

To validate that a delivered product (bespoke only), be it in-house or third party, meets the overall business and operational requirements as documented in the Pathway Baseline Functional Specification [8], supporting Product Description and EPIDs.

A1.3.2 Activities

Produce Product Acceptance Test Strategy for the release;

Test analyse the business and operational requirements (Requirements Catalogue), along with the relevant Product Description and EPIDs for a discrete product (be it developed in-house or by a third party). Produce a series of High Level Test Plans (HLTPs) which verify that the product has internal integrity, as well as correctly manages its APIs. Tests must be cross referenced back to the Requirements Catalogue;

Review the Physical Design documentation and progress the logical HLTPs to their physical state, producing Low Level Test Scripts (LLTSs): these include definitions of the input data states, run instructions and expected results. Tests should also include cross references back to the design products, allowing auditability and traceability;

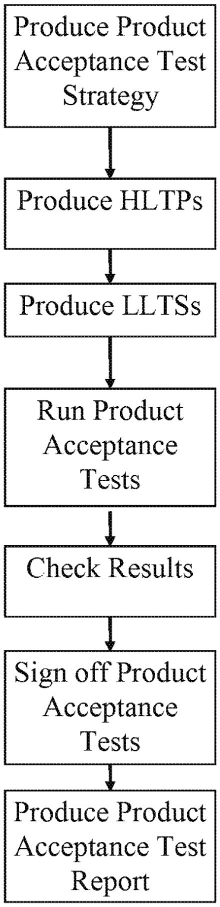
Run Product Acceptance Tests (PATs);

Check results from PATs against expected results;

Sign off PATs;

Produce Product Acceptance Test Report for the release.

A1.3.3 Product Acceptance Test Activities Dependency Diagram


Produce Product Acceptance Test Strategy
Produce HLTPs
Produce LLTSs
Run Product Acceptance Tests
Check Results
Sign off Product Acceptance Tests
Produce Product Acceptance Test Report

A1.4 SYSTEM TEST

A1.4.1 Purpose

To confirm that, once integrated, the different products developed/bought meet the baselined functional requirements.

To start to integrate the application software components with the supporting infrastructure software and gain early exposure re simple recovery and resilience issues.

To prepare the ground for Integration Testing.

A1.4.2 Activities

Produce System Test Strategy for the release;

Working with business representatives and Systems analysts, define and document a set of Business Threads: these are a high level, logical statement of business flow through the services. A mixture of threads will be required - some specific to a discrete service (e.g. Benefit Payments, running through CMS, PMS, Middleware, counter services); others running multiple integrated activities (e.g. Benefit Payments, with POCL service such as sending parcels, buying stamps etc.);

Using the Business Threads as an input, test analyse the business and operational requirements for the full set of products which constitute this release (using the Requirements Catalogue, Product Descriptions and EPIDs), producing a set of High Level Test Plans (HLTPs). These expand the threads into greater detail, and also extend the functional coverage. The HLTPs provide traceability back to entries in the Requirements Catalogue, as well as indicate any sequencing dependencies within each thread. The business representatives will be asked to review and sign off the HLTPs;

Produce Inter-system threads which validate the interfaces with other systems (i.e. outside Pathway domain, for example to CAPS, POCL Systems (e.g. FAD), British Gas systems, etc.);

Produce the Test Execution Plan (TEP), which defines how to configure the Low Level Test Scripts (LLTSs), ensuring that dependencies are satisfied but using the different functional components within the HLTPs to provide a broad coverage of functional permutations.

Migrate the HLTPs to a physical level, producing the Low Level Test Scripts (LLTSs). LLTSs include definitions of input data, run instructions and expected results. Tests should also include cross references back to the design products, allowing auditability and traceability;

Produce the Infrastructure thread which will build the necessary data infrastructure to support system test activities: this will include any migration activities required;

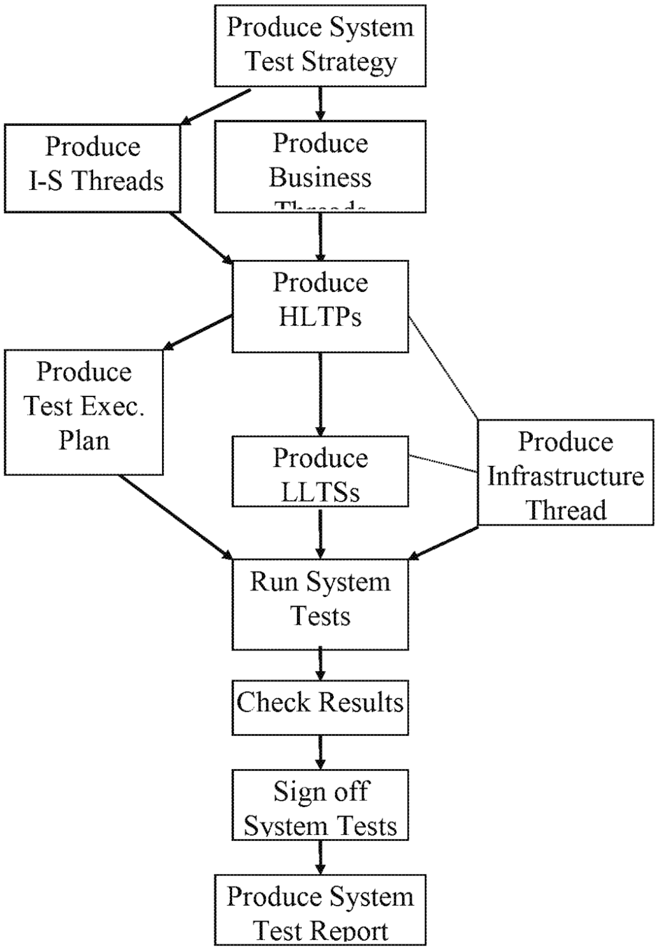
Run System Tests, using the information contained in the LLTSs combined with the Test Execution Plan to drive the combination of tests;

Check results against the expected results;

Sign off tests;

Produce System Test Report for the release.

A1.4.3 System Test Activities Dependency Diagram



A1.5 INTEGRATION TEST

A1.5.1 Purpose

To confirm that the different products, be they developed in-house or bought, fully integrate, meet the baselined functional and operational requirements and operate in a live shape environment.

To prove the end to end architecture of the system solution, and in particular with regard to resilience and recovery.

To confirm the performance attributes of the overall system and so verify that the SLAs and the NFRs from the TED [6] are met.

A1.5.2 Activities

Produce Integration Test strategy for the release;

Test analyse the Service Level Agreements and Non-Functional Requirements for this system taking supplementary detailed information from the technical specification and the performance budgets.

Review the Threads produced for System Testing, and identify whether any of the test requirements identified from the test analysis activity on SLAs and NFRs can be satisfied by running the thread tests;

Produce specific Integration HLTPs to ensure full coverage of the SLAs and NFRs: these HLTPs will cross reference the SLA and NFR products. In addition, cross references are also provided between Thread tests being used to provide full coverage of SLAs and NFRs. The business representatives will be asked to review and sign off the HLTPs;

Produce the Test Execution Plan (TEP), using the System Test HLTPs as well as the Integration HLTPs, defining how to configure the Integration Low Level Test Scripts (LLTSs). This activity ensures that dependencies are satisfied, with the added flexibility of using the different functional components within the HLTPs to provide a broad coverage of functional permutations.

Migrate the Integration HLTPs to a physical level, producing the Integration Low Level Test Scripts (LLTSs). LLTSs include definitions of input data, run instructions and expected results. Tests should also include cross references back to the design products, allowing auditability and traceability;

Upgrade the Infrastructure thread supporting System Testing to ensure that the data infrastructure required to support Integration Testing is established: this will include any migration activities required, as well as creation of the necessary data infrastructure on existing systems (e.g. CAPs);

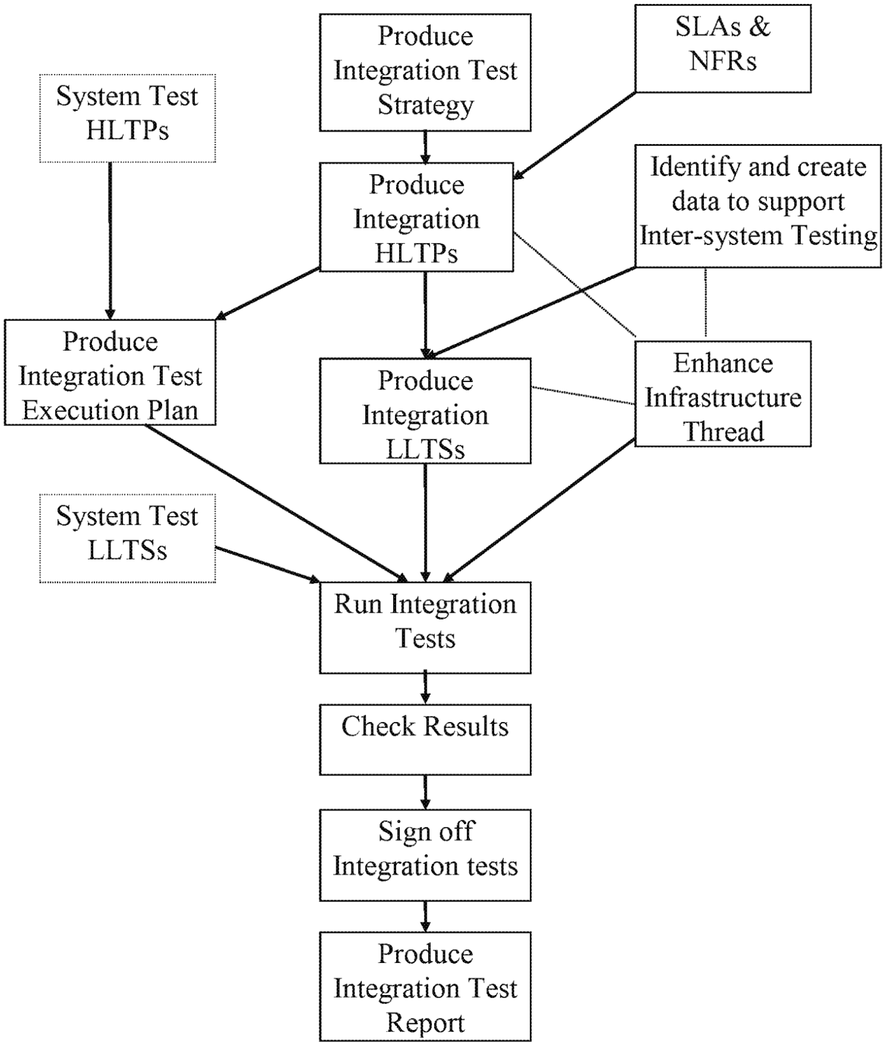
Run Integration Tests (which will include re-running of the selected System tests) using the information contained in the LLTSs combined with the Test Execution Plan to drive the combination of tests;

Check results against the expected results;

Sign off tests;

Produce Integration Test Report.

A1.5.3 Integration Test Activities Dependency Diagram


Produce Integration Test Strategy
SLAs & NFRs
System Test HLTPs
Produce Integration HLTPs
Identify and create data to support Inter-system Testing
Produce Integration Test Execution Plan
Produce Integration LLTSs
Enhance Infrastructure Thread
System Test LLTSs
Run Integration Tests
Check Results
Sign off Integration tests
Produce Integration Test Report

APPENDIX 2.

GLOSSARY OF TERMS

Term	Meaning
Acceptance Criteria	The document(s) agreed between the Customer and the Service Provider which lay down the criteria against which the system will be measured in the User Acceptance Trial when judging whether the system is to be formally accepted by the Customer.
Architectural Conformance	The testing lifecycle wherein the test objectives are oriented toward demonstrating that the system and its components conform to their architectural specification, including aspects such as structure, performance, security, operability, resilience, recovery,and other NFRs.
BA/POCL	The Customer
Black Box Testing	Testing of a system or its components which is planned and conducted without knowledge of the inner workings, looking only at the input, the specification of the system, and the output.
Business Integrity	The testing lifecycle wherein the test objectives are oriented toward demonstrating that the overall system satisfies the business need and when used in context and together with the associated business products, retains overall business integrity.
Business Procedures	The document(s) defining the business activity surrounding and controlling the use by the Customer and their staff of the computer system.
CM-BOM	The structured information set (document or database) held by CM which unambiguously defines the content of the system in terms of deliverables known as Configuarble Items, their inter-relationships, and their target locations and runtime attributes. Used as a blueprint of the system to guide and to audit the assembly of the overall system.
Demonstrator Phase	The bid phase defined by the Customer which demonstrates proof of concept of the proposal. It precedes the 'Operational Trial Phase'.
EPIDs	Unambiguous definitions of the physical nature of an interface between the system under test and an external system.
Functional Conformance	The testing lifecycle wherein the test objectives are oriented toward demonstrating that the system and its components conform to their functional specification.
HCI style	The standards by which a system is written to interface with people. These cover for example the structure, layout, format, colour, fonts, highlights and other conventions used on a

	computer input/output screen.

(continued)

Operating Instructions	The document(s) defining the operational control of the system, such as batch schedule control, recovery control, etc.
Operational Trial Phase	The main development and testing phase as defined by the Customer, following the ‘Demonstrator Phase’.
Output Handling Equipment	Ancillary equipment associated with the computer system or its output, which extends the service offered. For example, equipment used in conjunction with computer printers, to say collate and envelope and frank output targeted for posting.
Pathway	A consortium of companies acting as the Service Provider.
Perfomance Budgets	Targets set by the Technical Manager and derived from the SLAs, which specify the required performance characteristics of the system and its components.
Regression Testing	The process of demonstrating that a system and its components has not regressed to a ‘worse’ state folowing a change (usually to the software).
Rollout	The process of packaging and distributing the system out to the various locations at which it will operate and be used.
Service Level Agreement	The document(s) agreed between the Customer and the Service Provider which lay down the minimum operational criteria to which the service must be run and maintained.
Service Provider	Pathway - providing the service of development, maintenance and operational running of the BA/POCL system, according to agreed requirements and Service Level Agreements.
Supplier	one of the suppliers of Pathway - a 3rd party providing goods or services to Pathway, or one its direct sub-contractors working to specification by and on behalf of Pathway.
The Customer	BA/POCL
Trusted Status	the state a system or its components have reached on successful completion of a particular stage of testing - e.g. ‘System Test Trusted Status’following successful completion of System Test.
User Acceptance Trial	A trial of the whole system, following successful completion of all formal testing, and prior to release of the system to live, conducted by or on behalf of the Customer to inform the acceptance process. In the case of the Pathway system for BA/POCL this equates to the ‘Mock Trial’ referenced in the original proposal.
White Box Testing	Testing of a system or its components which exploits knowledge of or interrogates the inner workings.

END OF DOCUMENT