FSA Modernization Partner

NSLDS II Reengineering Test Plan

Version 1.0

July 19, 2002

Table of Contents

1 INTRODUCTION 3

1.1 BACKGROUND 3
1.2 TEST OBJECTIVES 3
1.3 DOCUMENT OBJECTIVES 3
1.4 TEST PROCESS 4
1.4.1 Develop Approach 4
1.4.2 Plan Test 4
1.4.3 Prepare Test 4
1.4.4 Execute Test 5
1.4.5 Manage Test 5
1.4.6 Establish and Operate Test Environment 5
1.5 CHANGE CONTROL PROCESS 5

2 TEST SCOPE 7

2.1 SECTION OVERVIEW 7
2.2 TEST ITEMS 7
2.2.1 High-Level Business Requirements 7
2.2.2 Detailed System Requirements 7
2.2.3 Test Items - Online Screens 7
2.2.4 Test Items - Interfaces 9
2.2.5 Test Items – EAI 13
2.2.6 Test Items - Reports 13
2.3 ASSUMPTIONS 16
2.3.1 Global Assumptions 16
2.3.2 Intersystem Test (IST) Assumptions 17
2.3.3 User Acceptance Test (UAT) Assumptions 17
2.4 RISKS AND CONTINGENCIES 17
2.5 RESOURCES AND RESPONSIBILITIES 18

3 TEST APPROACH 19

3.1 SECTION OVERVIEW 19
3.2 METHODOLOGY 19
3.2.1 The V-Model Approach 19
3.2.2 Phase Containment 20
3.2.3 Entry and Exit Criteria 21
3.3 TEST PHASES 21
3.3.1 Unit Test 21
3.3.2 Integration Test 22
3.3.3 System Test 23
3.3.4 Intersystem Test 24
3.3.5 Performance Test 25
3.3.6 User Acceptance Test 26
3.4 TEST MODEL 27
3.4.1 Test Cycles 27

3.4.2 Test Condition, Scenario, and Script Traceability to Requirements 28
3.4.3 Test Passes 28
3.5 SYSTEM INVESTIGATION REQUEST 29
3.5.1 SIR Severity 29
3.5.2 SIR Status 30
3.5.3 Re-test SIR Fixes 30
3.6 TEST DATA 30

4 DEVELOPMENT, TEST, AND PRODUCTION ENVIRONMENTS 32

5 TEST SCHEDULE 33

Document Control

Version Number	Description	Release Date	Author
1.0	Initial Issue	July 19, 2002	Jason Patton

1 Introduction

1.1 Background

The National Student Loan Data System (NSLDS) was established as part of the Higher Education Act of 1965, as amended, to provide a comprehensive repository of information about Title IV recipients and their loans, grants, lenders, guaranty agencies, servicers, and schools. Currently, NSLDS is hampered by a number of challenges related to discrepancies between the quality and timeliness of data feeds and the system of record, and its operating costs.

Given these challenges, a project to modernize the system – NSLDS Reengineering – has been undertaken to improve financial and data integrity, reduce operational costs, and improve customer satisfaction. At the heart of the design for a reengineered NSLDS solution are five "big ideas" for radically changing the underlying processes, data structures, and technical platforms supporting the existing system:

- Data Warehousing
- Internal FSA (Federal Student Aid) Direct Access
- Outsourced Enrollment Tracking
- Financial Partner Data Feed Reengineering
- Common Record Extension

The first phase of the NSLDS Reengineering effort is called NSLDS II. NSLDS II Release 1 will focus on the Data Warehousing and Internal FSA Direct Access opportunities, as well as assessing ways to support existing requirements through NSLDS II or other modernized systems. Later phases or releases of work will focus on the remaining three ideas.

This Test Plan covers the activities required to test and obtain acceptance for the NSLDS II Release 1 solution.

1.2 Test Objectives

The objectives of the NSLDS II Release 1 test effort are to:

- Verify the solution satisfies the business requirements and supports the NSLDS II business processes.
- Verify the functionality of the solution performs as designed.
- Verify the solution meets performance requirements.

1.3 Document Objectives

This Test Plan for the NSLDS II Release 1 solution supports the following objectives:

- Detail the activities required to prepare for and conduct the test effort.
- Define the scope of the test effort, including the items to be tested.
- Document the overall testing approach.
- Define the test environments required to conduct the test.

- Identify and communicate to responsible parties the tasks they are to perform and the schedule to be followed in performing these tasks.

1.4 Test Process

The process used to structure the test activities is illustrated in the diagram below and explained in the following sections. The NSLDS II Modernization Partner test team will carry out this process.

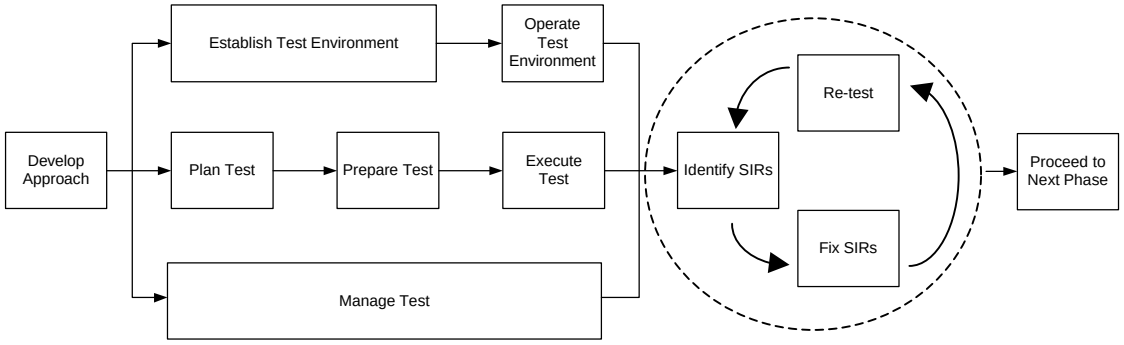

Figure 1, Test Process

1.4.1 Develop Approach

The *Develop Approach* phase consists of the following tasks:

- Define the test approach and test plan
- Identify the test phases
- Define the scope and objectives of the test phases
- Identify the phase containment strategy and entry and exit criteria
- Identify risks associated with the test effort

1.4.2 Plan Test

The *Plan Test* phase consists of the following tasks:

- Develop the test schedule
- Identify test conditions
- Identify test scenarios

1.4.3 Prepare Test

The *Prepare Test* phase consists of the following tasks:

- Define expected results
- Create test scripts
- Define test data
- Verify that the conditions, scenarios, scripts, and data validate functional and technical requirements

1.4.4 Execute Test

The *Execute Test* phase consists of the following tasks:

- Execute test scripts for each phase
- Verify expected results
- Log System Investigation Requests (SIRs)
- Update conditions, scenarios, scripts, and data with discrepancies
- Re-test SIR fixes

1.4.5 Manage Test

The *Manage Test* phase consists of the following tasks:

- Manage issues
- Facilitate / coordinate test activities
- Track progress against the test schedule
- Report status

1.4.6 Establish and Operate Test Environment

The *Establish and Operate Test Environment* phases consist of the following tasks:

- Define the test environment requirements
- Allocate environment resources (both human and physical assets)
- Perform configuration management activities

1.5 Change Control Process

During the system lifecycle process, the controlled baseline requirements may require updates and the Change Control Process is designed to manage these modifications. Changes may be required for a variety of reasons, including the addition of new technology or functionality, the resolution of issues, and in response to technical and operational tests and evaluations.

The change process for the NSLDS II Release 1 effort is illustrated in the following diagram. This process will be closely followed during the project's Construction (design, build, and test) phase. Identified changes will be logged as SIRs and discussed by the Change Control Board (CCB). Changes that are related to existing requirements will be assigned for remediation. Changes that result in a new requirement will be noted as an enhancement and will either be postponed to a future release or will be negotiated with the FSA Project Sponsor for inclusion into the current release.

Please reference the project Configuration Management Plan for additional information on the Change Control Process and the CCB.

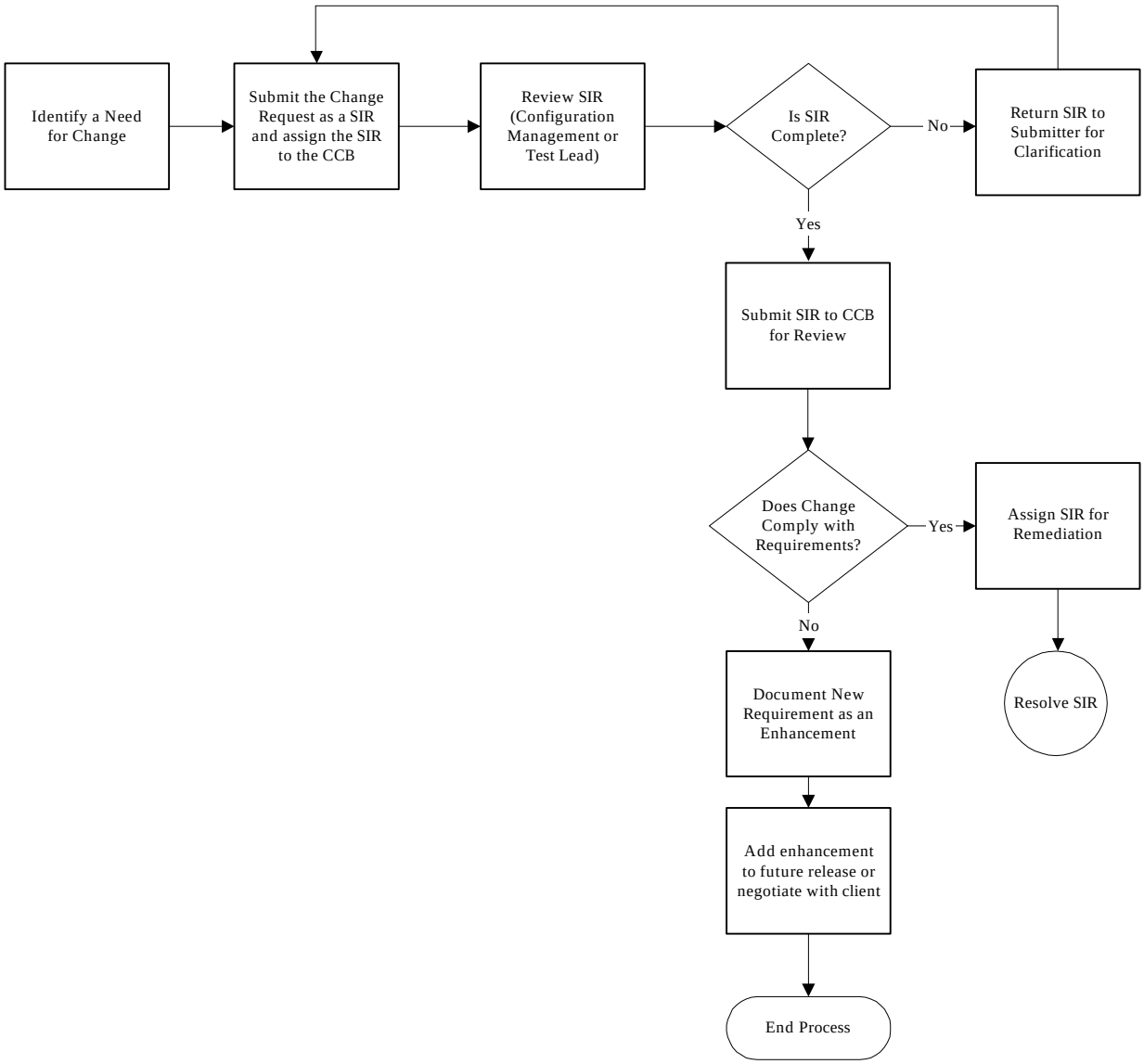

Figure 2, Change Control Process

2 Test Scope

2.1 Section Overview

This section will document the items to be tested, while identifying the assumptions and risks associated with the NSLDS II Release 1 test effort.

2.2 Test Items

The items in this section reflect the NSLDS II functionality identified during Preliminary Design. The requirements, screens, interfaces, and reports listed in the following sections are subject to change when the final Detailed Design deliverable is completed in the fall of 2002. The scope of the NSLDS II Release 1 test effort will be finalized at that time.

2.2.1 High-Level Business Requirements

Testing for NSLDS II Release 1 will consist of the following high-level business requirements:

- Student Aid Eligibility
- Financial Aid History
- Student Transfer Monitoring
- Loan Transfer Tracking
- Financial Partners (FP) Payment / Payment Reasonability
- Cohort Default Rate Calculation and Distribution
- Enrollment Tracking / Reporting
- Audit and Program Review
- Research and Policy Development
- Budget Formulation and Execution

2.2.2 Detailed System Requirements

Please reference the Detailed Requirements contained within the NSLDS II Preliminary Design: System Requirements – Revised document for the NSLDS II system requirements.

2.2.3 Test Items - Online Screens

The online screens for NSLDS II Release 1 are contained within either the Financial Aid Professional Web Site or the Student Access Financial Aid Review Web Site.

2.2.3.1 Financial Aid Professional Web Site

The following Financial Aid Professional Web Site screens will be tested for Release 1:

- System Log On
- Privacy Act
- Menu
- Change Password
- System Requirements
- Contact Us
- Frequently Asked Questions (FAQ)

- Download Help
- Help
- Message Detail
- Security
- Loan History
- Student/PLUS Borrower Name Search
- Borrower Name History
- Borrower SSN History
- Loan Detail
- Overpayment History List
- Overpayment Add
- Overpayment Add and Student Add
- Overpayment Display
- Overpayment Update
- Overpayment Delete Confirmation
- Pell Grant History
- Student Access Interface
- Enrollment Summary
- Enrollment Detail
- Enrollment Timeline
- Enrollment Maintenance
- Enrollment Add
- Enrollment Update
- Enrollment Reporting Schedule
- Enrollment Reporting Schedule Create
- Enrollment Reporting Schedule Modify
- Enrollment Waiver
- Organization Contact List
- Organization Contact Detail
- Organization Contact Update
- Organization Contact Delete
- Organization Contact Add
- Organization Search
- Data Provider Schedule
- Repayment Information
- Cohort Default Rate History List
- Rate Substitution
- Rate Combination
- Suspend Default Rates
- School Profile
- Reporting Menu
- Shared Reports
- Transfer Monitoring List
- Student Monitoring Add

- Student Monitoring Detail
- Monitoring Alert Review
- School Transfer Profile
- School Transfer Profile Add
- School Transfer Profile Update

2.2.3.2 SAFAR Web Site

The following Student Access Financial Aid Review (SAFAR) Web Site screens will be tested for Release 1:

- FAQ
- Browser Information
- Browser Setup
- Download Browser
- Domestic Version Information
- System Requirements
- Gathering Your Information
- Contact Us
- Links
- PIN Request and Information
- Financial Aid Review
- Loan Detail
- Glossary
- Logoff

2.2.4 Test Items - Interfaces

The systems and data types that comprise the NSLDS II Release 1 interfaces are listed in the following table.

System	Interface Direction	Data Type	System Category
Schools	Inbound to NSLDS	Enrollment Data - SSCR Roster File	External
Schools	Outbound	Enrollment Data - SSCR Submittal File	External
Schools	Inbound	Enrollment Data - SSCR Load Process Error Notification File	External
Schools	Outbound	Enrollment Data - SSCR Load Process Correction File	External
Schools	Outbound	Loan Level Data - Perkins Submittal File	External
Schools	Inbound	Loan Level Data - Perkins Error File	External

System	Interface Direction	Data Type	System Category
Schools	Inbound	Loan Level Data - Perkins TEF File	External
Schools	Inbound	Loan Level Data - Perkins Error Submittal Notification File	External
Schools	Outbound	Loan Level Data - FAT Request File	External
Schools	Inbound	Loan Level Data - Fat Results File	External
Schools	Outbound	Enrollment Data - Student Transfer Monitoring Inform File	External
Schools	Inbound	Enrollment Data - Student Transfer Monitoring Results File	External
Schools	Inbound	CDR Notification Letters and LRDR	External
Clearinghouse	Inbound	Enrollment Data - SSCR Roster	External
Clearinghouse	Outbound	Enrollment Data - SSCR Submittal File	External
Clearinghouse	Inbound	Enrollment Data - SSCR Load Process Error Notification File	External
Clearinghouse	Outbound	Enrollment Data - SSCR Load Process Correction File	External
Clearinghouse	Outbound	Enrollment Data - Direct Loans	External
Guaranty Agency	Inbound	Enrollment Data - Student Status Change File	External
Guaranty Agency	Outbound	Loan Level Data - FFEL Submittal File	External
Guaranty Agency	Inbound	Loan Level Data - FFEL Load Process Error File	External
Guaranty Agency	Inbound	Loan Level Data - TEF File	External
Central Processing System (CPS)	Outbound	Enrollment - Pre-screening	Internal
Central Processing System (CPS)	Inbound	Enrollment - Pre-screening	Internal
Central Processing System (CPS)	Inbound	Enrollment - Post Screening	Internal
Central Processing System (CPS)	Outbound	Enrollment - Post Screening, Error File	Internal
Central Processing System (CPS)	Outbound	Student Demographics	Internal

System	Interface Direction	Data Type	System Category
Post-Secondary Educational Processing System (PEPS)(possible Consistent Answers Replacement)	Outbound	Enrollment - School-Data	Internal
Post-Secondary Educational Processing System (PEPS)(possible Consistent Answers Replacement)	Inbound	Enrollment - School-Data - Error File 2	Internal
Post-Secondary Educational Processing System (PEPS)(possible Consistent Answers Replacement)	Inbound	Loan - Cohort Default Rate	Internal
Post-Secondary Educational Processing System (PEPS)(possible Consistent Answers Replacement)	Inbound	Loan - Non-SAIG OPEID list	Internal
Post-Secondary Educational Processing System (PEPS)(possible Consistent Answers Replacement)	Outbound	Loan - Letter files	Internal
Direct Loan Servicing System (DLSS)	Outbound	Loan	Internal
Direct Loan Servicing System (DLSS)	Inbound	Enrollment - Status Update	Internal
Direct Loan Servicing System (DLSS)	Outbound	Exit Counseling Update Request (FFEL data)	Internal
Direct Loan Servicing System (DLSS)	Inbound	Exit Counseling Update Results (FFEL data)	Internal

System	Interface Direction	Data Type	System Category
Debt Collection System (DCS)	Outbound	Loan / Grant Data	Internal
Common Origination and Disbursement (COD) (Replacing RFMS)	Outbound	Loan - Pell Disbursement Data	Internal
Common Origination and Disbursement (COD) (Replacing RFMS)	Inbound	Loan - Pell Disbursement Data, Error File	Internal
Financial Management System (FMS)	Inbound	Loan - LPIF Summary	Internal
Financial Management System (FMS)	Inbound	Loan Summary AMF	Internal
Financial Management System (FMS)	Outbound	FFEL Lender Demographics	Internal
Financial Management System (FMS)	Outbound	Lender Special Allowance	Internal
Financial Management System (FMS)	Outbound	Lender Interest	Internal
Total Permanent Disability (TPD)	Outbound	Loan - TPD Loan Discharge Data	Internal
Ombudsman (Seibel)	Outbound	Student data request	Internal
Ombudsman (Seibel)	Inbound	Student Data results	Internal
Student Aid Internet Gateway	Outbound	Destination File	Internal
FP Datamart	Inbound	Lender Cohort Default Rates	Internal
FP Datamart	Inbound	Form 2000 Data	Internal
FP Datamart	Inbound	Default Claim Data	Internal
FP Datamart	Inbound	FFEL Loan Data	Internal
FP Datamart	Inbound	GA VFA Data	Internal

Table 1, Interface Inventory

2.2.5 Test Items – EAI

The following interfaces will connect to the NSLDS II system through the use of the EAI (Enterprise Application Integration) Bus:

- Common Origination and Disbursement (COD)
- Central Processing System (CPS)
- Direct Loan Servicing System (DLSS)
- Financial Management System (FMS)
- Financial Partners (FP) Datamart
- Ombudsman
- Post-Secondary Educational Processing System (PEPS)
- Student Aid Internet Gateway (SAIG)
- Total Permanent Disability (TPD)

EAI testing will focus on the movement of files from the source to the target system. Processing of those files will be handled within the appropriate test phases.

2.2.6 Test Items - Reports

The following list contains the reports that will be created to support NSLDS II:

- FFEL Loan Disbursements Report
- Agency Portfolio Status Report
- Lender Portfolio Status Report
- Loans Held by Lender Report
- Lender Loan Portfolio Report
- School Eligibility Report
- Loan Discharge Report
- Total Aid Volumes by School Report
- Awarded Federal Aid by Student Report
- Loan Recipient Listing by School Report
- Grant Recipient Listing by School Report
- School Pell Grant Volume Report
- Pell Grant Volume by State Report
- Lender Loan Volume by State Report
- Lender Loan Disbursement to School Summary Report
- Loans Guaranteed by GA Report
- Student Contact Information with Loan Summary Report
- Reconciliation File Report
- Error Detail File Report
- Rolled-up Loan Report
- Unpaid Underlying Loans (PF) Report
- Unpaid Underlying Loans (PF and CL) Report
- Subrogated Loans Report
- Outstanding Principal Balance Not Updated Report
- Open Loans Zero Balances Report

- Old Loans Held by GA Report
- Old Loans Held by Lender or Servicer Report
- Loans Open > 15 Years Report
- No Loan Disbursements Report
- Repurchased Loans Report
- Outstanding Principal Balance Twice Guaranty Amount Report
- Duplicate Loans Report
- GLOS "Samples" File Report
- No Provider ID Report
- Full Error Count Report
- Lender List Report
- Lender Totals Report
- Errors Not in NSLDS Report
- SSCR Detail Benchmark Report
- Forward-Moving Date Entered Repayment Report
- Sum of Disbursement + Cancellation > Guaranty Amount Report
- Potential LPIF Overpayment Report
- Insurance No Reinsurance Report
- Reinsurance No Insurance Report
- Closed Loans w/OPB > $0 Report
- “Unclassified” School Code Report
- Loans Status Codes Used Report
- Excess Loan Amounts Report
- Open Loans Zero Balances Report (Data Integrity Alert #1)
- Potential AMF Overpayment Report
- Recent Claims Report
- Defaulted Loans with Borrower Collections Report
- Moving Outstanding Principal Balance Report
- Old Loans Report
- FFEL School Volumes Report
- Direct Loan School Volumes Report
- Disbursement + Collection with no Insurance or Reinsurance Report
- Date Entered Repayment Report
- School Repayment Information Loan Detail Report
- Cohort Default Rate History Report
- Repayment History Summary Report
- CDR Preview Summary Report
- DM Summary Report
- Loan Forbearance and Deferment Report
- Participation Rate Index Appeal Report
- SSCR Transmittal Report
- Financial Aid History Report
- GA Loan Error Rates and Top 10 Errors Report
- Loan Profile Data by SSN Report

- Reasonability Quarterly Form 2000 Report
- Accounts Maintenance Fee Report
- Loan Processing and Issuance Fee Report
- Reasonability Quarterly Backup Form 2000 Report
- Reasonability Yearly Form 2000 Report
- Reasonability Annual Backup Form 2000 Report
- Federal Receivables Report
- Form 2000 GA Summary Computation Report
- Field Listing for Form 2000 Monthly Payment Reasonability Report
- Form 799 Quarterly Payment Reasonability Report
- School Overpayment Report
- Lender Loan Cancellation Report
- Exit Counseling Report
- Transfer Monitoring Summary Report
- Enrollment Reporting Historical Report
- Enrollment Status by School Report
- GA Submittals for Current Month Benchmark Report
- GA Submittal Monthly Error Benchmark Report
- GA Submittal Cumulative Error Benchmark Report
- GA Portfolio for Current Fiscal Year Benchmark Report
- GA Refunds for Fiscal Year Benchmark Report
- GA Portfolio for Previous Fiscal Year Benchmark Report
- GA Refunds for Previous Fiscal Year Benchmark Report
- GA Open Loan Data Benchmark Report
- GA Loans in Deferment Benchmark Report
- GA Loans in Default Benchmark Report
- GA Loan Status Summary Benchmark Report
- Current Loan Balance Benchmark Report
- Total Loan Guarantee Benchmark Report
- GA Portfolio Quarterly Benchmark Report
- GA Loan Refund Quarterly Benchmark Report
- GA Portfolio Historical Quarterly Activity Benchmark Report
- GA Loan Refund Historical Quarterly Activity Benchmark Report
- GA Open Loan Data Quarterly Benchmark Report
- Cancelled Direct Loans for Academic Year Benchmark Report
- Direct Loan Disbursements for Academic Year Benchmark Report
- Direct Loans for Academic Year Benchmark Report
- Direct Loan Refunds for Academic Year Benchmark Report
- Open Direct Loan Profile by Loan Type Benchmark Report
- Direct Loans with Zero Balance Benchmark Report
- Direct Loans in Deferment Benchmark Report
- Direct Loans in Grace Period Benchmark Report
- Direct Loan Cumulative Error Benchmark Report
- Direct Loan Monthly Error Benchmark Report

- Direct Loan Submittal Profile Benchmark Report
- Direct Loan Error Benchmark Report
- DCS Cumulative Error Benchmark Report
- DCS Loan Portfolio Benchmark Report
- DCS Loans with Zero Balances Benchmark Report
- Closed Perkins Loans Benchmark Report
- Perkins Loans in Excess Benchmark Report
- Open Perkins Loans Benchmark Report
- Perkins Portfolio Benchmark Report
- Perkins Submittal Benchmark Report
- Perkins Submission Grouped Error Benchmark Report
- Perkins Submittal Date of Birth Error Benchmark Report
- Perkins Duplicates By Date Benchmark Report
- Perkins Duplicates By Loan Type Benchmark Report
- Perkins Duplicates By School Code Benchmark Report
- Perkins Error Benchmark Report
- NSLDS II Content Benchmark Report

2.3 Assumptions

The following assumptions have been made with regard to the test effort that is outlined in this document. Any deviations from these assumptions may impact the testing schedule.

2.3.1 Global Assumptions

- Any System Investigation Request not captured in the requirements or design documentation will be documented as an enhancement and staged for a future release, unless the Modernization Partner and FSA agree upon its inclusion.
- The development, test, and production environments will be located at the FSA hosting site.
- The development, test, and production environments will be available 24 hours per day x 7 days per week with the exception of planned maintenance efforts.
- The development and test environments will mimic the production environment's hardware and software requirements.
- The environments at the FSA hosting site will be accessible by all project resources (Development and Test).
- Testers must have IDs with read and update access to the system. These IDs will allow testers to view contents of the database tables for verification of test results, and to manipulate the test data in the database in order to component test the modules.
- Test conditions will be defined and agreed upon within the Detailed Design phase.
- Exit criteria will be followed to advance from one test phase to the next.
- A common tool will be used to capture System Investigation Requests. This tool will be developed using Rational ClearQuest.
- Testing of the data conversion from the legacy NSLDS system to the NSLDS II data structure is documented in the Conversion and Migration Strategy.

- Testing of the legacy NSLDS mainframe solution will not be performed during NSLDS II Release 1. The production implementation of NSLDS II Release 1 will allow for the retirement of the mainframe.
- The Debt Collection System (DCS) is currently being looked at as part of FSA's Common Servicing for Borrowers (CSfB) initiative. Due to very recent changes in scope and timelines for efforts to reengineer DCS, its impact on NSLDS II is still uncertain. The NSLDS II team will work with CSfB project staff to determine potential interface impacts resulting from the eventual rollout of CSfB. As part of the first release of NSLDS II, submission of data weekly in tape format from DCS will be supported. Additional short-term initiatives to change the DCS to NSLDS II file transmission mechanism from tape to flat file over the EAI Bus will be investigated.

2.3.2 Intersystem Test (IST) Assumptions

- FSA will be responsible for organizing the internal and external participation for the intersystem test, and that the various participants remain on schedule.
- The intersystem test will only test limited transactions with the internal and external interface systems.
- The internal and external interface systems are responsible for creating the test data they send to NSLDS II, based on specifications provided by the Modernization Partner team.
- The intersystem test will require that fixes to System Investigation Requests are re-tested using local copies of the original input files.
- Connectivity to the NSLDS II environments will be enabled for the internal and external interface systems during the IST for the transfer and receipt of test files.
- The intersystem test will require that files are transferred between systems using communication protocols resembling those that will exist in production for NSLDS II.

2.3.3 User Acceptance Test (UAT) Assumptions

- UAT will be based on written, mutually agreed-upon scripts, which will be provided to FSA for review and feedback and sign-off. This effort will provide expected results, repeatable testing conditions, and defined exit criteria.
- User groups will be available to execute testing in accordance with the approved schedule.
- FSA will be responsible for organizing the internal and external participation for the user acceptance tests.

2.4 Risks and Contingencies

The risks that have the potential to hamper the progress of this testing effort that are currently identified are located in the table below.

Risk	Contingency
Testers must have IDs with read and update access to the system. These IDs will allow testers to view contents of the database tables for verification of test	Members of the development team with IDs will provide assistance if access cannot be granted to the necessary users.

Risk	Contingency
results, and to manipulate the test data in the database in order to component test the modules.	
Coordination of intersystem testing activities between multiple groups and application systems.	Identify a central FSA point of contact and a specific point of contact for the interface system to assist with the coordination of testing between the appropriate participants. FSA will be responsible for organizing the internal and external participation for the intersystem test.
Availability of FSA and external resources for UAT.	FSA and the Modernization Partner will prepare a schedule for UAT that will outline the required testing days and times. FSA will ensure that the user groups will be available to execute testing in accordance with the approved schedule.

Table 3, Risks and Contingencies

2.5 Resources and Responsibilities

The NSLDS II Modernization Partner development team will prepare and execute the test scripts for the unit and integration test phases.

The Modernization Partner test team will prepare and execute the test scripts for the system and performance tests. The test team will also aid in the facilitation of the intersystem and user acceptance testing, including the preparation of test scripts.

The NSLDS II internal and external interface systems, outlined in **Section 2.2.4 Test Items - Interfaces**, will assist the Modernization Partner test team with the execution of the intersystem test.

The following user groups will be included in the execution of test scripts for the user acceptance test:

- FSA
- ED
- Schools
- Clearinghouse
- Guaranty Agencies

FSA will assist with the facilitation of the interface system participation for the intersystem test. In addition, FSA will provide available internal resources and assist with obtaining external resources for the user acceptance test.

The Assistive Technology Group within the FSA Technology Center will conduct the Section 508 review.

3 Test Approach

3.1 Section Overview

This section will discuss the methodology, test phases, test model, issue logging procedures, and test data that will be used to test the items that were identified in **Section 2.2 Test Items**.

3.2 Methodology

There are three key pieces to the testing methodology that will be implemented during the NSLDS II Release 1 test effort. They are the V-Model approach, phase containment, and entry and exit criteria.

3.2.1 The V-Model Approach

The following diagram displays the software development V-Model approach used by the NSLDS II Modernization Partner team.

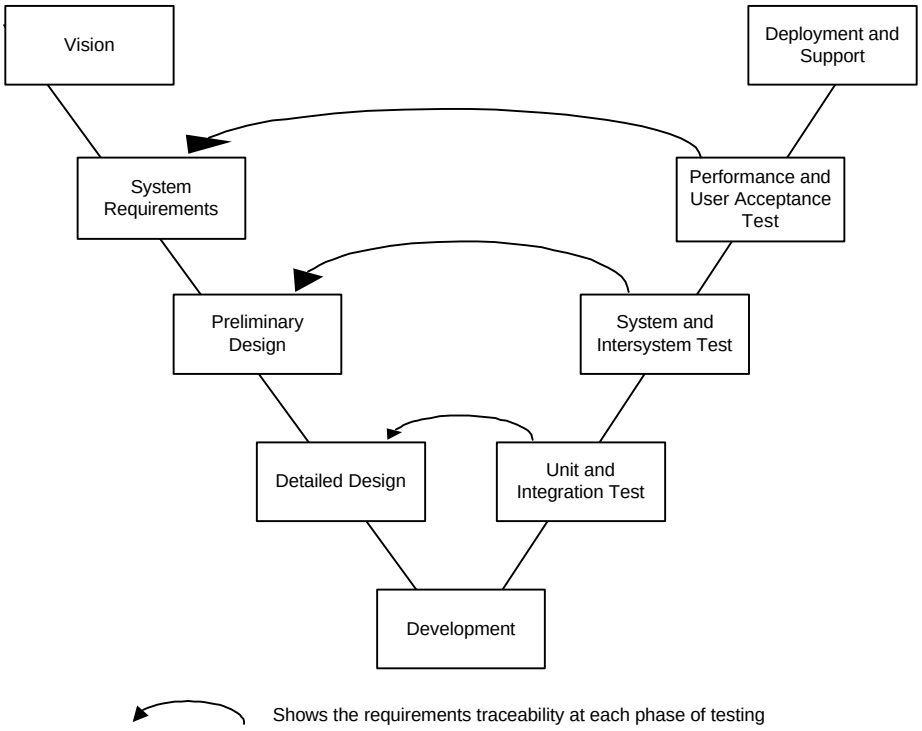

Figure 3, NSLDS II Software Development V-Model Approach

The V-Model provides a structured software development framework, emphasizing building quality into the project starting in the initial requirements phase and continuing through the final testing phase.

The V-Model will serve as the standard approach that will be used to test the Release 1 functionality. The left side of the V-Model contains the requirements and design phases, the bottom of the model contains the Development phase, and the right side contains the software test phases.

The requirements and design side of the V-model contains the phases that must be successfully completed prior to development. These phases are: Vision, System Requirements, Preliminary Design, and Detailed Design. As a software development project moves down the requirements and design side of the model, the application is designed in greater detail with each phase until the design is complete and development begins.

The testing side of the V-Model contains the test phases that a system must successfully complete before it may be deployed to production. The NSLDS II Release 1 test effort will contain six test phases: Unit, Integration, System, Intersystem, Performance, and User Acceptance. According to the V-Model, the level of detail involved in each test phase decreases as the software development project progresses up the right side of the model. As the design phases became more detailed with each passing phase, the test phases begin at the most detailed level and progress to a higher level of detail.

The arrows on the diagram above display the requirements traceability at each phase of testing. The unit and integration test phases will test the components depicted in the Detailed Design document. The system and intersystem test phases will focus on the items from the Preliminary Design documentation, while the performance and user acceptance test phases will test the high-level business processes and scenarios found in the System Requirements documentation.

By implementing the V-Model approach, the following benefits will be obtained:

- Improved quality and reliability
- Reduction in the amount of rework
- Reduction in the cost of problem correction
- Efficient testing by focusing on the objectives of the various tests
- Requirements traceability allowing informed scope decisions
- Improved risk management
- Higher probability of delivery on schedule

In order to achieve these benefits, the processes of phase containment and entry and exit criteria must be implemented along with the V-Model. The phase containment, entry and exit criteria, and the V-Model processes work together to enhance the quality of a deliverable or release.

3.2.2 Phase Containment

Phase containment is used to verify quality deliverables are passed between project phases. The goals of phase containment are to minimize the number and severity of errors passed from one phase into the next and to enable issues to be identified earlier in the testing process, generally resulting in an easier and quicker error resolution process.

To encourage phase containment, progress to the next phase of testing will not occur until a determined exit criteria threshold is achieved.

3.2.3 Entry and Exit Criteria

Phase containment is fostered through the use of entry and exit criteria. For each test phase, entry and exit threshold criteria have been established. Entry criteria are standards used to determine when a software development project is ready to enter a given phase, while exit criteria are standards used to understand what is required of a software development project to exit a particular phase.

The entry and exit threshold criteria for relevant test phases will be reviewed before terminating a test phase and proceeding into the next one. Any deliverable failing to meet its criteria will be returned to its current phase.

3.2.3.1 Entry Criteria

The entry criteria for each phase of the NSLDS II Release 1 test effort are:

- Completion of the relevant Design phase
- Developed components completed and inventoried
- Test preparation activities for the particular test phase are complete
- Resources are available to execute tests
- Test environment and test data are in place
- Successful completion of the prior V-Model phase

3.2.3.2 Exit Criteria

Each test phase must complete the following items:

- Relevant test cases and conditions have been executed
- Identified issues have been properly documented and worked through the resolution process (within the limits specified in **Section 3.3 Test Phases**).

The exit criteria for each test phase are described in **Section 3.3 Test Phases**. If the number and severity of the open SIRs meets the exit criteria, the system will move to the next test phase and the remaining open SIRs will be resolved then. If the product has not met the exit criteria, SIRs will be resolved and tested within the current phase until the exit criteria are met.

3.3 Test Phases

A test phase is a set of related test activities that share the same objective. The NSLDS II Release 1 test effort will be divided into the following logical phases of testing: unit, integration, system, intersystem, performance, and user acceptance. Each phase is described in detail within the following sections.

3.3.1 Unit Test

A unit test is the test of an individual component, or module, of the solution. The objective of a unit test is to verify that the component correctly implements the designed specifications.

Unit test scripts are based on knowledge of how the logical unit is designed to work and will be written once the detailed design specifications are completed. The test conditions for the unit test are unique to this effort and will not be repeated in other efforts once the exit criteria have been achieved. Unit test scripts will include tests for field ranges, values and lengths, functions, data validation, data dependencies, and any special processing contained in the module. The unit test scripts will be updated if changes occur to the requirements and/or the detailed design specifications.

Modules will be developed and become available for testing independently of other modules. The developer responsible for the module will perform the unit test and identify and document the errors related to the independent operation of the program. Level 1 and 2 SIRs will be remediated and closed prior to progressing to integration test. In addition, there should be no more than an average of three open Level 3 SIRs per module. No defined limit will be set for Level 4 SIRs while Level 5 SIRs are system enhancements and will not count towards the exit criteria for this phase.

The exit criteria for the unit test is defined as having equal to or less than the following quantity of open SIRs:

Qty	SIR Level
0	1 – Critical
0	2 – Major
Average of 3 per module	3 - Average
No defined limit	4 – Minor
NA	5 – Enhancement

Table 4, Unit Test Exit Criteria

3.3.2 Integration Test

Following the completion of development and unit testing, integration testing will be performed on sets, or groups, of dependent modules. An integration test tests the compatibility of related components to verify that they function properly when integrated.

While the unit test scripts focus on the smallest logical unit of code, the integration test scripts focus on how multiple components work together and communicate. The integration test scripts contain scenarios to test the interactions between system components as they are assembled to form progressively larger subsystems. By the end of the integration test, each function will be integrated into the system and internal NSLDS II interfaces will be tested.

The development team will perform the integration test. At the end of the integration test, relevant issues should be identified and documented. Level 1 and 2 SIRs will be remediated

and closed prior to progressing to system test. A determination on the number of open Level 3 and 4 SIRs will be at the discretion of the Modernization Partner (MP) delivery manager. Level 5 SIRs are system enhancements and will not count towards the exit criteria for this phase.

The exit criteria for the integration test is defined as having equal to or less than the following quantity of open SIRs:

Qty	SIR Level
0	1 – Critical
0	2 – Major
TBD - MP	3 - Average
TBD - MP	4 – Minor
NA	5 – Enhancement

Table 5, Integration Test Exit Criteria

3.3.3 System Test

The system test phase will contain two groups: System Application Testing and Section 508 Testing.

At the end of the system test, relevant issues should be identified and documented. Level 1 and 2 SIRs will be remediated and closed prior to progressing to the user acceptance test. A determination on the number of open Level 3 and 4 SIRs will be at the discretion of the Modernization Partner delivery manager. Level 5 SIRs are system enhancements and will not count towards the exit criteria for this phase.

The exit criteria for the system test is defined as having equal to or less than the following quantity of open SIRs:

Qty	SIR Level
0	1 – Critical
0	2 – Major
TBD - MP	3 - Average
TBD - MP	4 – Minor
NA	5 – Enhancement

Table 6, System Test Exit Criteria

3.3.3.1 System Application Testing

System application testing concentrates on requirements and business processes, and ensures that the system requirements have been met. System testing will test the components and interfaces working together as a whole, integrated solution.

The Modernization Partner test team will handle the creation and execution of system test scripts. The test scripts for this phase will be developed using the high-level scenarios developed during Preliminary Design.

Ideally, system testing cannot begin until all modules have passed both the unit and integration test phases because the modules must be fully integrated in the test environment for system testing. However system testing may begin prior to the acceptance of all modules within the prior phases, but the impacted modules must be moved to the end of the test phase. This decision will be at the discretion of the Modernization Partner delivery manager.

3.3.3.2 Section 508 Testing

An accessibility review will be conducted during the system test phase by the FSA Technology Center (within the Office of the Chief Information Officer). The purpose of the review is to verify that persons with disabilities may perform the necessary processes of the NSLDS II system.

This review is mandated by Section 508 of the Rehabilitation Act of 1998. Section 508 requires that electronic and information technology developed, procured, maintained, or used by Federal government agencies must be accessible to persons with disabilities. The system must allow a person with a disability to have comparable access to and use of information as a person without a disability.

Section 508 Testing will be conducted on both the front-end screens and reports. The ED Assistive Technology Group must issue a Web Site Accessibility Review with a grade of “PASS” prior to the production deployment of NSLDS II Release 1.

3.3.4 Intersystem Test

Intersystem, or end-to-end, testing will test the NSLDS II interfaces with both the internal and external interface systems. The objectives of the intersystem system are to:

- Test the NSLDS II system to confirm that it can receive transmissions from the interface systems and process the associated transactions.
- Test with the internal and external interface systems to verify they can receive transmissions from NSLDS II and can process the associated transactions.
- Test the infrastructure between interface systems, focusing on the ability for the NSLDS II system to send and receive data using the communication protocols that will exist in production.

The interfaces discussed within **Section 2.2.4 Test Items - Interfaces** will be the focus of this test. Not all transactions will be tested with the interface systems. The interface systems will be

responsible for creating the test data they send to NSLDS II, based on specifications provided by the Modernization Partner team.

The functionality tested during the intersystem test will also be tested during the system test phase. During system test, the test team will create and stage the incoming files based upon the technical specifications, and outgoing files will be generated by NSLDS II but not transmitted. In addition, local testing of the EAI interfaces and other communication protocols will be performed. Therefore intersystem testing is independent of the other test phases, but should occur within an integrated environment and preferably after at least one complete pass (test and SIR remediation) of system test. Intersystem testing may start concurrent with the second system test pass and may continue into the user acceptance phase.

At the end of the intersystem test, relevant issues should be identified and documented. Level 1 and 2 SIRs will be remediated and closed prior to the production deployment of NSLDS II Release 1. Level 3 SIRs will be assigned for pre- or post-production remediation by the CCB, while Level 4 SIRs will be at the discretion of the Modernization Partner delivery manager. Level 5 SIRs are system enhancements and will not count towards the exit criteria for this phase.

The exit criteria for the intersystem test is defined as having equal to or less than the following quantity of open SIRs:

Qty	SIR Level
0	1 – Critical
0	2 – Major
TBD - CCB	3 - Average
TBD - MP	4 – Minor
NA	5 – Enhancement

Table 7, Intersystem Test Exit Criteria

3.3.5 Performance Test

The performance test phase will verify that the system can handle the expected workload by testing the system's business processes and business-driven scenarios and applying increased stress to the environment. The performance goals will be taken from the performance requirements established during the Detailed Design phase (*i.e.* the ability to handle a database with greater than one terabyte of data).

Since the performance test focuses on the performance goals of the system and not functionality, it is independent of the other test phases. However, the performance test must occur within an integrated environment and preferably after at least one complete pass (test and SIR remediation) of system test. Therefore, the performance test may start concurrent with the second system test pass and may continue into the user acceptance phase.

The Modernization Partner test team will handle the creation and execution of performance test scripts. The performance test scripts will expand on the system's performance requirements. These scripts will detail the steps necessary to verify that the deployed solution will handle the expected workload while performing standard business functions.

At the end of the performance test, relevant issues should be identified and documented. Level 1 and 2 SIRs will be remediated and closed prior to the production deployment of NSLDS II Release 1. Open Level 3 SIRs will be assigned for pre- or post-production remediation by the CCB, while Level 4 SIRs will be at the discretion of the Modernization Partner delivery manager. Level 5 SIRs are system enhancements and will not count towards the exit criteria for this phase.

The exit criteria for the performance test is defined as having equal to or less than the following quantity of open SIRs:

Qty	SIR Level
0	1 – Critical
0	2 – Major
TBD - CCB	3 – Average
TBD - MP	4 – Minor
NA	5 – Enhancement

Table 8, Performance Test Exit Criteria

3.3.6 User Acceptance Test

The user acceptance test (UAT) phase will focus on the system's business processes and scenarios, with a purpose of verifying that the modules can interact to handle the core business processes performed. The UAT also tests the usability of the new system and procedures, and facilitates an understanding of the technology and the business change being implemented.

User acceptance test scripts will be prepared by the NSLDS II Modernization Partner test team and will expand on the system's requirements. User acceptance scripts test the requirements and business processes from a user perspective. They include enough test conditions to determine if the application meets the user acceptance criteria, but do not include steps for all test conditions. The UAT tests the system at a high level, with more detailed testing occurring during the prior test phases.

The User Groups will run these pre-defined scripts with assistance from the NSLDS II Modernization Partner test team. All testing will be based on written, mutually agreed-upon scripts, which will be provided to FSA for review and feedback. This effort will provide expected results, repeatable testing conditions, and defined exit criteria.

At the end of the user acceptance test, relevant issues should be identified and documented. Level 1 and 2 SIRs will be remediated and closed prior to the production deployment of NSLDS II Release 1, while Level 3 and 4 SIRs will be assigned for pre- or post-production remediation by the CCB. Level 5 SIRs are system enhancements and will not count towards the exit criteria for this phase.

The exit criteria for the user acceptance test is defined as having equal to or less than the following quantity of open SIRs:

Qty	SIR Level
0	1 – Critical
0	2 – Major
TBD – CCB	3 – Average
TBD – CCB	4 – Minor
NA	5 – Enhancement

Table 9, User Acceptance Test Exit Criteria

3.4 Test Model

3.4.1 Test Cycles

The following three test cycles will be performed for NSLDS II:

- Cycle 1 - Normal Horizontal Processing
- Cycle 2 - Normal Vertical Processing
- Cycle 3 - Error Processing

3.4.1.1 Cycle 1 - Normal Horizontal Processing

The purpose of Normal Horizontal Processing cycle is to discover any software deficiencies that would occur in a typical end to end walk through of the system functionality. This cycle may be either script based or scenario based, and tests the system at a high level with an emphasis on integrated modules.

This cycle will be utilized for the system, intersystem, performance, and user acceptance test phases.

Certain Cycle 1 processing scripts (from either the system or user acceptance test phases) will be identified for "critical functionality testing". These scripts will be used for any system regression testing that is required. These scripts will test to verify that critical business functions are ready for deployment.

3.4.1.2 Cycle 2 - Normal Vertical Processing

The Normal Vertical, or Detailed, Processing cycle consists of a detailed series of component tests that will thoroughly test functional areas. This cycle may be script based or scenario

based, and the tests are independent and allow for concurrent testing. SIRs found within this cycle may not be as critical as those found during Cycle 1, but could disrupt the system processing.

This cycle will be utilized for the unit and integration test phases.

By the completion of Cycles 1 and 2, the system should be technically sound and data flow through the system should be correct. A successful execution of Cycles 1 and 2 will verify that transactions, database updates, etc. are functioning correctly.

3.4.1.3 Cycle 3 - Error Processing

This cycle focuses on verifying and validating extreme scenarios that may disrupt normal processing and push the software to extremes in an effort to reveal breaks within the logic that may hinder the functionality under certain conditions. This cycle consists of scripts designed to produce environmental errors pertaining to a specific object or feature in the software, such as record navigation and field length/validation.

This cycle will be utilized during the unit and performance test phases.

3.4.2 Test Condition, Scenario, and Script Traceability to Requirements

Test conditions will be defined during the Detailed Design phase. These conditions will be written using the detailed requirements from the Preliminary Design phase. The test conditions will then be mapped to the related detailed requirement.

Conditions will be grouped into scenarios. The scenarios will contain a related set of conditions, with the condition mapped to the detailed requirements.

Finally, test scripts will be prepared during the Development phase that define the steps, input data, and expected output to test the scenarios. A single scenario may be split across multiple test scripts, or multiple scenarios may be included within a single test script.

3.4.3 Test Passes

A test pass is defined as one full test execution cycle. Development remediation activities are conducted between passes within the same test phase. Additional passes will be run within a test phase until the test meets the defined exit criteria.

Test passes will be performed in steps. A pass contains the following steps:

1. The tester executes the test scripts.
2. The tester enters a SIR when results from the test deviate from what is expected.
3. The test lead notifies the development lead when the pass is complete.
4. The development lead runs the SIR report at the end of the pass.
5. The development lead performs an impact analysis on the SIR list and reports his/her findings to the test lead.
6. SIR fixes are completed and moved into the testing environment according to Configuration Management guidelines.

7. The development team updates the SIR database with the SIR status information.
8. The development lead notifies the test lead that testing may resume.
9. The test team begins a new pass and continues this process until the exit criteria have been achieved for the phase.

If a critical error occurs within a test pass that prevents the entire set of scripts from being executed and the fix requires lengthy remediation, the pass will be considered complete and the SIR resolution phase will begin. Once the SIR fixes are migrated into the test environment, a new pass will begin. The beginning of a new pass will be at the discretion of the test lead, and a new pass may begin with unresolved SIRs outstanding. Test passes will continue until the scripts for that pass have been executed and the exit criteria have been satisfied. The amount of time necessary to complete a test pass will vary depending on the particular area of testing being performed.

3.5 System Investigation Request

Whenever a test condition or test script step does not return the expected result, the tester will log the discrepancy between the expected and actual result as a SIR. SIRs will be logged and tracked in a central repository, or SIR database. The SIR database will include a long and short description of the issue (including the actual and expected results), its severity and status, the affected test area or module, the test phase, pass, and tester.

3.5.1 SIR Severity

The following table contains the SIR severities and definitions. The tester will use these definitions to provide an initial severity, but the Modernization Partner delivery manager and the CCB will have the authority to make a final decision on SIR severity levels.

ID	Label	Description
1	**Critical**	The anomaly results in the failure of the complete software system, a subsystem, or a software module within the system.
2	**Major**	The anomaly results in a failure of the complete software system, a subsystem, or a software module within the system. There is no way to make the failed component(s) work; however there is an acceptable processing alternative that will achieve the desired results (an acceptable work around exists).
3	**Average**	The anomaly does not result in a failure, but causes the system to produce incorrect, incomplete, or inconsistent results, or the anomaly impairs system usability.
4	**Minor**	The anomaly does not cause a failure, does not impair usability, and the desired processing results are easily obtained by working around the anomaly. The anomaly may also be the result of non-conformance to a standard or related to the aesthetics of the system.

ID	Label	Description
5	**Enhancement**	The anomaly is a request for an enhancement. Anomalies at this level may be deferred to a future release or will be treated as an addition to the defined project scope if the Modernization Partner and FSA agree upon its inclusion.

Table 10, SIR Severity Levels

3.5.2 SIR Status

The following table contains the SIR Status levels that will be used during this effort and a definition of each:

SIR Status	Definition
Opened	The SIR was created and is awaiting assignment or review
In-Progress	The SIR was assigned and that resource is currently working on its resolution
Resolved	Corrective action was completed and the SIR is awaiting validation
Closed	Corrective action has been completed and validated
Postponed	The CCB agreed to postpone the corrective action on this SIR to a later release
Duplicate	The issue identified in this SIR is identical to another open SIR, and the resolution to the duplicate SIR will be tracked through the original open SIR

Table 11, SIR Statuses

3.5.3 Re-test SIR Fixes

As SIR fixes are completed, the script in which the SIR was encountered will be re-tested. These SIRs may have originated from an earlier pass, the same pass, or from another script that has one or more components in common. The test lead will coordinate with the development lead to group related issues together, thus minimizing the delay of executing the next pass of the test script.

SIR fixes will be re-tested using local copies of the original input files, data, etc. After conducting a re-test, SIRs that fail are reopened and newly identified SIRs are logged in the SIR database. Additional passes will continue to be conducted until the exit criteria are achieved.

3.6 Test Data

Three types of test data are required for NSLDS II testing. They are:

1. Development Database → The development database will be used for the unit and integration tests. The development database will be created by manually loading a small set of sample test data into the NSLDS II warehouse database structure.
2. Partial Production Data Extract → A partial extract of production data will be obtained from the legacy NSLDS production mainframe database. The extract will be converted

into the NSLDS II warehouse database format and will be used for the system, intersystem, and user acceptance test phases.

3. Full Production Data Extract → A full extract of production data will be obtained from the legacy NSLDS production mainframe database. The extract will be converted into the NSLDS II warehouse database format during conversion system testing and will be used for performance testing.

4 Development, Test, and Production Environments

The following diagram depicts the environments that will be required for the NSLDS II project.

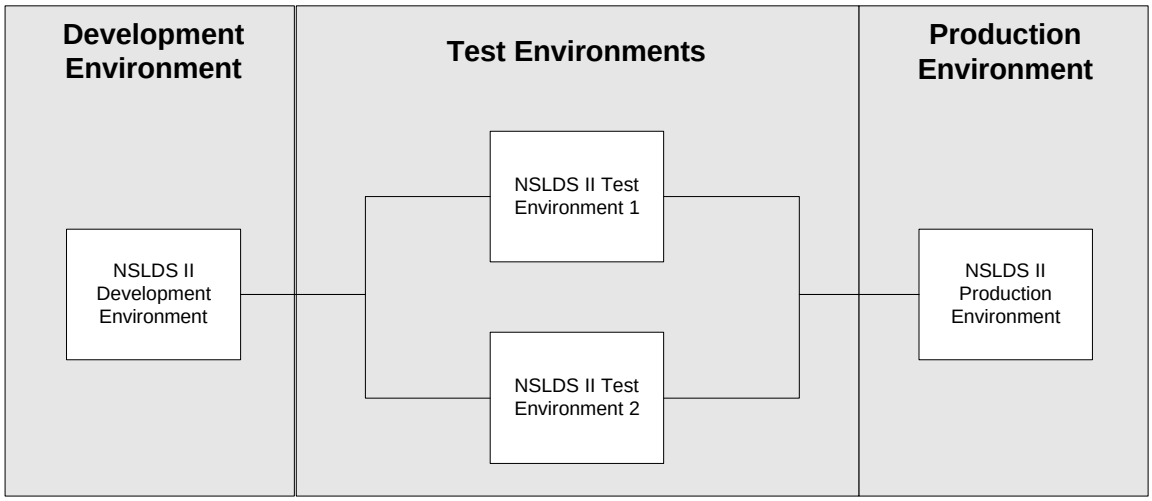

Figure 4, NSLDS II Development, Test, and Production Environments

The following five environments will serve different purposes:

- NSLDS II Development Environment – This environment will be created using the data warehouse architecture for NSLDS II and will be used for development, unit testing, and integration testing. The environment will mirror the production environment, except the database used will be smaller in scale.
- NSLDS II Test Environment 1 – Test environment 1 will be created using the data warehouse architecture for NSLDS II and will be used for system, intersystem, and user acceptance testing. This environment will mirror the production environment, except the database used will be scaled to handle the converted legacy NSLDS partial extract.
- NSLDS II Test Environment 2 - Test environment 2 will be created using the data warehouse architecture for NSLDS II and will be used for system, intersystem, and user acceptance testing. This environment will mirror the production environment, including a database with similar capacity (~2 Terabytes).
- NSLDS II Production Environment - This environment will be created using the data warehouse architecture for NSLDS II and will be used for conversion and performance testing during the Construction project phase and used as the NSLDS II production environment for the Deployment and Support project phases. The database in this environment will be scaled to handle the converted data set from the legacy NSLDS production database.

The host and architecture of these environments will be finalized during the Detailed Design phase.

5 Test Schedule

The following figure represents a high level organization of the test activities and dependencies.

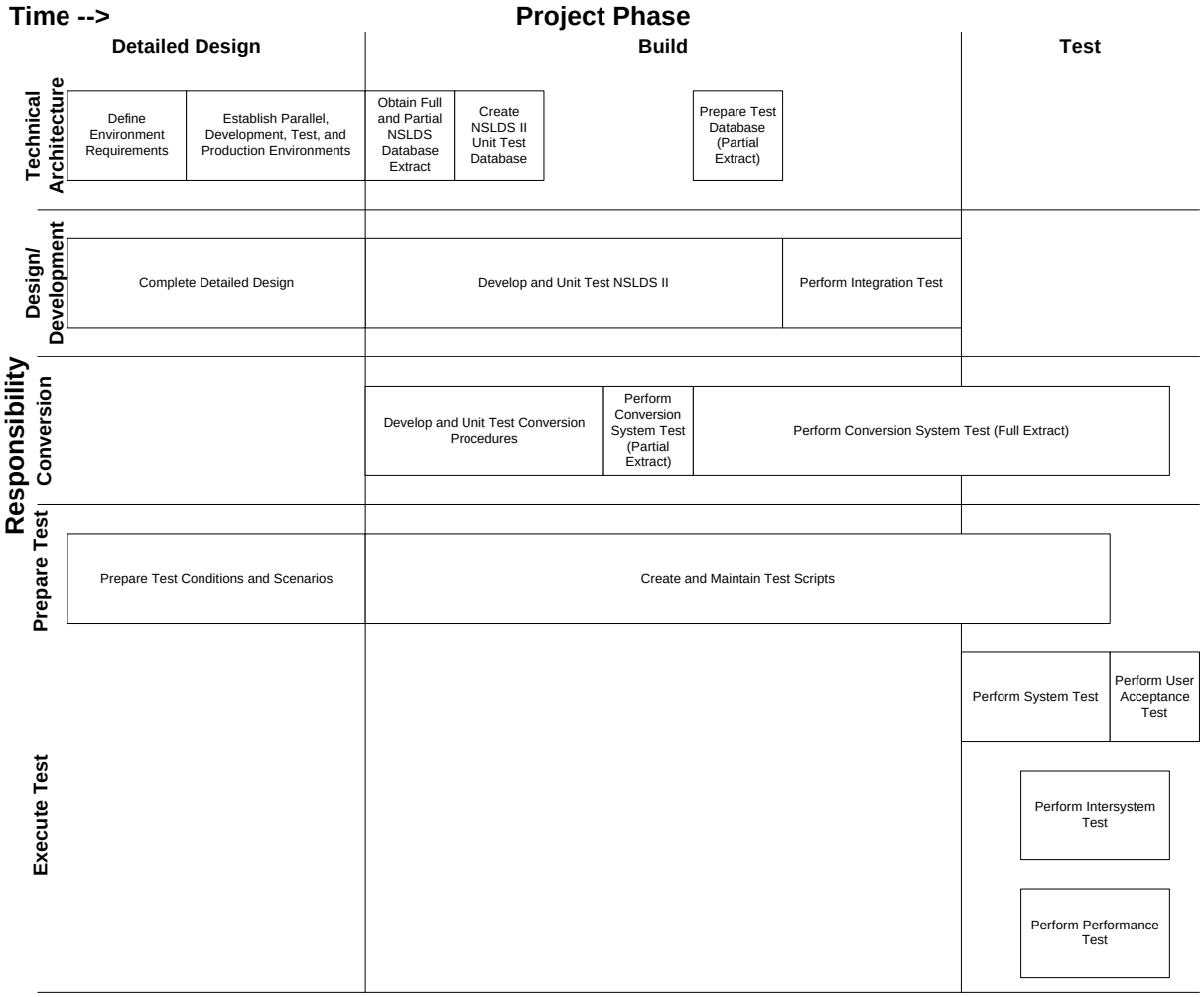

Figure 5, High Level Test Schedule

The overall project schedule is maintained using Microsoft Project. The detailed test schedule will be prepared during the Build phase.