

# MSS Modernization Project

## Test and Evaluation Master Plan (TEMP)

April 7, 2023

Version 0.1

Status: DRAFT

Authorized for issue by:

<Name>

NSP Program Manager

\_\_\_\_\_

Signature

\_\_\_\_\_

Date

<Name>

Unisys Corporation

Program Manager

\_\_\_\_\_

Signature

\_\_\_\_\_

Date

# Document Revision History

Version	Date	Author	Summary of Changes
0.1	April 7, 2023	Jeff Corn	Initial Draft Document

# Table of Contents

Document Revision History .....	i
1. Introduction .....	1
1.1 Purpose .....	1
1.2 Objectives .....	1
1.3 Scope .....	2
1.4 Risk and Assumptions .....	3
1.4.1 Risks .....	3
1.4.2 Assumptions .....	3
1.5 Roles and Responsibilities .....	4
2. Types of Testing .....	6
2.1 Testing Workflow .....	7
3. Test Script Development Process .....	8
3.1 Gap Analysis (GAP Documentation and RTM) .....	8
3.2 Gap Documentation Specification .....	8
3.2.1 Requirements Traceability Matrix (RTM) .....	9
4. Management of Testing .....	10
4.1 Tracking and Correcting System Defects .....	10
4.2 Record Collection and Maintenance .....	10
4.3 Test Script Pass/Fail Criteria .....	11
4.4 Defect Classification (Priority) .....	11
4.5 Defect Correction and Verification .....	12
4.6 Regression Testing .....	12
4.7 Test Cycles .....	12
5. Entry and Exit Criteria .....	14
5.1 Test Plan .....	14

5.1.1	Test Plan Entry Criteria .....	14
5.1.2	Test Plan Exit Criteria.....	14

## List of Figures

<b>Figure 2-1. Testing Workflow.</b> .....	7
--	---

## List of Tables

<b>Table 1.5-1. MSS Test Program Roles and Responsibilities.</b> .....	4
<b>Table 2-2. Types of Testing.</b> .....	6
<b>Table 4.4-3. Defect Severity Classifications.</b> .....	11
<b>Table 4.7-4. Order of Testing.</b> .....	13

# 1. Introduction

## 1.1 Purpose

The purpose of this Test Plan for the Message Switching System (MSS) Modernization Project is to address the overall objectives, strategy, approach, scope, and methodology for performing MSS solution testing. The testing will start at the lowest system components and move up through final user testing. The levels of testing that will be addressed by the Test Plan include:

- **Unit Testing** – Internal testing conducted by the Unisys Team to verify individual software components. Approved internally by Unisys.
- **System Testing** – Internal testing conducted by the Unisys Team to verify individual software components are operating together. Approved internally by Unisys.
- **Factory Acceptance Testing** – Internal testing conducted by the Unisys Team with NSP to ensure that the system capabilities are available and work in a factory setting, and that the documentation associated with the Solution reflects the design and is usable.
- **System Integration/Acceptance Testing** – External testing conducted by the Unisys Team to verify that software components are operating with agency integration points. The NSP will approve System Integration Testing results and will serve as an indication the system is ready for User Acceptance Testing (UAT).
- **User Acceptance Testing** – External testing conducted by NSP and end user staff to verify the system is meeting the administrative, business and operational requirements of the complete system. The system will be tested against the itemized requirements listed in the GAP documents and the Requirements Traceability Matrix (RTM). UAT results will be approved by the NSP.

## 1.2 Objectives

The objective of the Test Plan is to experience the system through each level of development to identify system problems, software defects and deviations from the stated business requirements. Each objective provides a different context to the test effort and alters the way in which testing should be approached. The objective for the evaluation effort in the Test Plan includes the following:

- Identify as many bugs as possible.
- Find important system problems.

## MSS Test and Evaluation Master Plan (TEMP)

- Assess perceived quality risks.
- Advise about perceived release delivery risks.
- Verify requirements.
- Advise about product quality.
- Fulfill contract requirements and mandates.

The MSS Modernization Project Test Plan will be used by MSS Project Management Office (PMO) to govern and direct the details the testing policies and procedures. The Test Plan will provide visibility to stakeholders in the testing effort that adequate consideration has been given to various aspects of the overall testing effort.

- Identify the tests performed during each Development Iteration.
- Describe the test approach for Unit, System, Factory Acceptance, System Integration and User Acceptance Testing.
- List criteria for test entrance, exit, suspension, and resumption for System, Factory Acceptance, System Integration, and User Acceptance Testing.
- List test deliverables.
- Describe testing workflow.
- Describe environmental needs for each level of testing.
- Define roles and responsibilities for the testing activities and clearly delineate which tasks are to be performed by Unisys and which are to be performed by the NSP staff.
- Summarize management process and procedures.

The Test Plan will be updated (if needed) for each of the project development iterations.

### 1.3 Scope

The Test Plan define the policies, procedures, and responsibilities for the MSS software releases. The test activities are based on the component GAP Assessment documents and RTM that will encompass the following:

- Functional Operations per each Iteration
- Administrative Operations
- Integration Operations to external systems
- Security Operations (Roles/Permissions).

## MSS Test and Evaluation Master Plan (TEMP)

Test activities will be performed using a combination of test tools and test scripts that include the following:

- System User Interfaces - User interfaces allowing user input to the system
- Server system and performance monitoring
- Server event viewer
- SQL Server Management Console.

### 1.4 Risk and Assumptions

#### 1.4.1 Risks

- On time availability and reliability of servers
- On time availability of the test and production external justice systems with which the MSS system interfaces
- Late delivery of code may affect ability to complete test execution on schedule
- The testing environment, if not available or incorrectly configured, may affect ability to complete test execution on schedule
- Reliance on limited resources with extensive domain and application knowledge may affect ability to complete test execution on schedule.

#### 1.4.2 Assumptions

Prior to performing any System, Factory Acceptance, System Integration, or User Acceptance Testing, it is assumed that all of the following are true:

- Unisys will internally perform and approve Unit and System testing before conducting Factory and System Integration Testing with the NSP
- All design documents and the RTM have been baselined and approved by the NSP
- All requirements have been baselined and approved by the NSP
- All code/objects/software has completed, properly configured and successfully installed on the test servers by the Unisys Team
- Unisys Team will be responsible for cataloging and reporting system defects

## 1.5 Roles and Responsibilities

The Unisys Team has overall responsibility for the test effort’s success including quality and test advocacy, resource planning and management, and resolution of issues that impede the test effort. The Unisys Team will identify and define the required tests, monitor detailed testing progress and results in each test cycle, and evaluate the overall quality experienced as a result of testing activities. The Unisys Test Manager will be responsible for coordinating the overall test effort. The role involves identifying the appropriate techniques, tools, and guidelines to implement the required tests and to give guidance on the corresponding resource requirements for the test effort. The Unisys Test Manager will be responsible for updating the Test Plan as each development iteration is reached or as changes become necessary.

**Table 1.5-1** identifies the roles and responsibilities for the NCJIS Test Program:

**Table 1.5-1. MSS Test Program Roles and Responsibilities.**

Name/Title	Role	Responsibility
<Name> <Phone> <Email>	NSP Project Manager	<ul style="list-style-type: none"> <li>Approve NCJIS Project Test Plan</li> <li>Approve NCJIS Project Test Cases and Test Scripts</li> <li>Coordinate Integration Testing (external agencies) and User Acceptance Testing efforts</li> <li>Approve NCJIS Factory Acceptance, System Integration, and User Acceptance Testing Results</li> </ul>
<Name> <Phone> <Email>	Unisys Team Project Manager	<ul style="list-style-type: none"> <li>Coordinate testing activities with Unisys Test Manager</li> <li>Conduct internal review of all test material before submitting to the Nevada DPS Project Manager</li> <li>Review and report the status of NCJIS defect log</li> <li>Manage NCJIS User Acceptance Testing efforts</li> </ul>
<Name> <Phone> <Email>	Unisys Team Test Manager	<ul style="list-style-type: none"> <li>Develop Test Plans</li> <li>Develop and revise Test Plan</li> <li>Develop and execute test cases</li> <li>Generate Factory Acceptance, System Integration, and User Acceptance Test scripts and procedures</li> <li>Create any other Test Deliverables</li> <li>Enter test defects in defect repository</li> <li>Provide system test reporting during testing efforts</li> <li>Coordinate review meetings with Component Team Lead(s)</li> </ul>
<Name> <Phone> <Email>	Unisys Team Component Team Lead	<ul style="list-style-type: none"> <li>Manage development and implementation of internal Unit and System testing</li> <li>Assign defects to software developers</li> <li>Monitor and verify defect resolution</li> <li>Review software defect list</li> </ul>
<Name> <Phone> <Email>	NSP Test Lead	<ul style="list-style-type: none"> <li>Coordinate User Acceptance Testing</li> <li>Conduct User Acceptance Testing</li> <li>Report system defects to the Unisys Test Manager</li> <li>Verify software bugs fixes</li> </ul>



MSS Test and Evaluation Master Plan (TEMP)

Name/Title	Role	Responsibility
		<ul style="list-style-type: none"> <li>Report testing progress to the Nevada DPS Project Manager</li> </ul>
<Name> <Phone> <Email>	End Users	<ul style="list-style-type: none"> <li>Conduct User Acceptance Testing</li> <li>Report system defects to the Nevada DPS Test Lead</li> <li>Verify software bugs fixes</li> </ul>

## 2. Types of Testing

Table 2-1 defines each type/level of testing that will be performed throughout the iterations through final User Acceptance Testing (UAT).

**Table 2-2. Types of Testing.**

Testing Type	Responsibility	Overview	Testing Artifacts	Test Execution Environment
Unit Testing	Unisys Team	Internal testing conducted by Unisys Team to verify individual software components are operating according to MSS project specifications. Unit Testing will occur during each development iteration of the project.	Unisys Team will develop Unit test scripts based on the function of each discreet software objects (e.g. code, scripts, configurations, and components). The Unit test scripts will be designed to exercise each atomic software object/component and verify it is working in accordance with the system requirements.	Developer station (laptop, tower, etc.)  Development Environment
System Testing	Unisys Team	Internal testing conducted by Unisys Team to verify individual software components are operating together to provide MSS project functionality according to specifications. System Testing will occur during each development iteration of the project.	Unisys Team will create System tests scripts that will combine the capabilities of various Unit test scripts. The System tests scripts will perform a general application function that will be traceable to specific requirements defined in the RTM and/or GAP Documents.	Test Environment
Factory Acceptance Testing (FAT)	Unisys Team	Internal testing conducted by Unisys Team with NSP team to verify the system capabilities are available and work in a factory setting.	Unisys Team will create FAT test scripts to verify baseline functionality within each component and any updates (GAP) made to the system per MSS project requirements.	Test Environment
System Integration Testing	Unisys Team will conduct tests in concert with NSP.	System Integration Testing will occur during each development iteration of the project. Successful system integration testing will signify that the system is ready for user acceptance testing for that delivered functionality.	Unisys will create System Integration test scripts that will combine the capabilities of various system test scripts and the capabilities of external agency integration points to meet the MSS system operational requirements.	Test Environment
User Acceptance Testing (UAT)	Unisys team will create the UAT test scripts.  NSP will conduct the testing and report results.	Successful User Acceptance Testing will signify that the MSS solution is ready to enter production.	Unisys will create User Acceptance Testing scripts by revising the system integration test scripts (where needed) to reflect the final operational requirements of the MSS system.	Test Environment

MSS Test and Evaluation Master Plan (TEMP)

Testing Type	Responsibility	Overview	Testing Artifacts	Test Execution Environment
	NSP will be responsible for approving the User Acceptance Test Results.			

## 2.1 Testing Workflow

Figure 2-1 illustrates the testing processes.

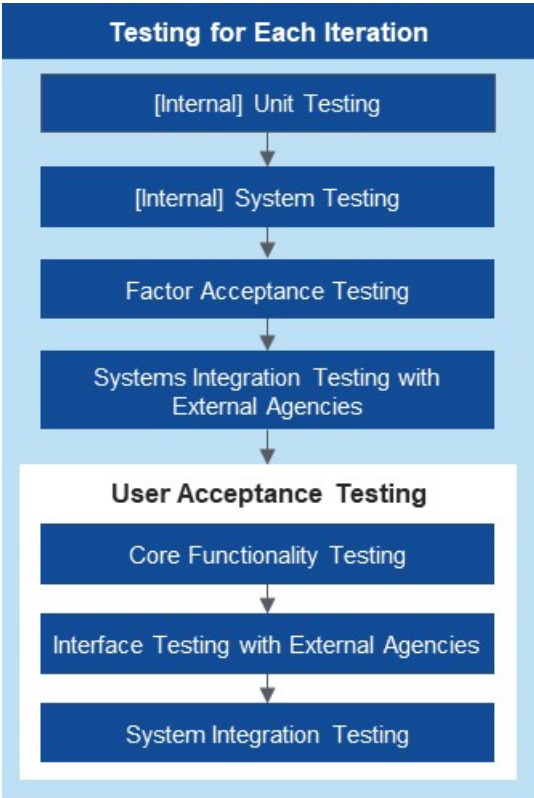


Figure 2-1. Testing Workflow.

## 3. Test Script Development Process

The development of test scripts is an iterative building block process where business requirements are translated into low level technical requirements that are used to develop discrete software objects (e.g. code, scripts, configurations and components) that will be combined to satisfy the business requirements the MSS system. The low-level software objects are tested individually through Unit test scripts. Software objects will be combined and reused to form more complex system capabilities that will be tested through System test scripts. Factory acceptance testing will be a step to verify baseline functionality and configuration of the system before moving into system integration testing. The progression will continue as external system capabilities are tested through System Integration test scripts. Finally, User Acceptance test scripts will be developed by transforming the System Integration test scripts into tests and operations that will be performed by the end user.

### 3.1 Gap Analysis (GAP Documentation and RTM)

The MSS system requirements and capabilities are defined through the GAP Analysis consisting of deviations in existing vendor product functionality that are specific to MSS project specification requirements. The GAP Document is used to define the operational flow of systems, while the RTM is a catalog (listing) of the system's requirements based on functions and features. Taken together, the GAP Document and the RTM provide developers, testers, and project sponsors with a powerful mechanism to define system requirements and provide a basis to test against. Once the GAP Analysis has been finalized and approved by the NSP, Unisys will begin to develop the software objects required to meet the specifications of the MSS system. Unisys Team will use the requirements documented in the GAP Analysis as the basis for developing test scripts, processes, and procedures.

### 3.2 Gap Documentation Specification

A GAP Document defines the system behavior deviations from the baselined functionality. It defines (as necessary) the screen flows, workflow modifications, specific component specifications, reports modifications, and security requirements.

A set of requirements is considered baselined when the GAP Document containing the set of requirements has been approved by the NSP Project Manager. Unisys will update the baseline document for each of the subsequent development iterations. The NSP project manager will be responsible for approving the baseline GAP Documents and updates for subsequent iterations.

### 3.2.1 Requirements Traceability Matrix (RTM)

The RTM is an itemized catalog of system requirements, functions, and features. The RTM will be used as the master reference to perform User Acceptance Testing. A set of requirements is considered baselined when the RTM containing the set of requirements has been approved by the NSP Project Manager. Unisys will update the baseline document for each of the subsequent development iterations, and the NSP project manager will be responsible for approving these baselines accordingly.

## 4. Management of Testing

Unisys will create and manage a test register and a test log. The test register will be a master inventory of all test scripts. It will list all test scripts and identify each phase, system, test type, and associated requirement that applies to the test script. Unisys will update the test register with any additions, modifications, or deletions of the data in a test script as well as additions, modifications, or deletions of test scripts. The test log will organize scripts for a particular test type. The test log will list all scripts that are to be performed in the activity for a test type for a specific system in each implementation phase. Unisys will use the test log to document the test results and any required follow-up, including defect reporting, defect correction, and regression testing.

### 4.1 Tracking and Correcting System Defects

The test scripts for each development iteration will represent the scope of testing that is conducted during the specific test type(s). Unisys and NSP will use the execution of the test scripts to determine whether defects exist in the MSS system. A defect is a flaw or an imperfection in a deliverable that causes its failure to conform to an approved systems requirement defined in the GAP Analysis and requirements traceability matrix. The defect classification will be documented in the Test Log and will be assigned pass/fail status for individual test scripts.

### 4.2 Record Collection and Maintenance

In addition to the test scripts, additional documents will be used during testing to track the tests and results, especially defects and corrections, including:

- **Test register.** The Unisys Test Manager will create a test register that lists all test scripts and identifies each phase, system, and test type that is associated with each test script.
- **Test log.** The Unisys Test Manager will create a test log that lists all scripts to be performed for each phase, system, and test type. Unisys will use the log to document the test number; test date; test results (pass/fail); and any required follow-up, including defect reporting, defect correction, and regression testing.
- **Defect report form.** A defect report will be completed for each defect identified during testing. The tester who identifies the defect will initiate the defect report. Basic information will be reported, and the defect report will be given to the Unisys Test Manager for tracking and management.
- **Defect register.** The Unisys Test Manager will create a defect register on which all reported faults will be recorded. The Unisys Test Manager will use the defect register to track the date opened, status, any corrective actions, owner, and date resolved.

### 4.3 Test Script Pass/Fail Criteria

The criteria for determining each test script’s success or failure will be specified in the script. There are four scenarios for determining each test script’s success or failure:

- **Pass.** If the test script indicated an expected response and the response was observed, the test passed.
- **Pass with unrepeatable error.** If the response specified in the test was observed during test executions, but the test failed once because of an unrepeatable error, the test will be considered passed. An unrepeatable error is defined as an error that was observed but cannot be repeated and analyzed by the project team to develop a proper repair. A defect report will be completed and tracked for reference in case the unrepeatable error recurs.
- **Pass with unrelated anomalies.** If the response specified in the test was observed, but a problem was detected in an area of functionality unrelated to the test script while the test script was executed, the test will be considered passed. The unrelated problem will be documented in a defect report and associated with the test script that will be used to subsequently verify the defect’s correction.
- **Fail.** If the test script indicated an expected response and the response was not observed, the test failed. If the execution of a test script provides an expected response other than the generation of one or more priority C defects, the script will be considered passed. A defect report will be completed for any defects that are identified.

### 4.4 Defect Classification (Priority)

The NSP will assign the defect classification in accordance with the classifications defined in the table below, to be verified and agreed to by the Unisys Team. Once assigned a classification and the organization responsible for investigation, resolution, or both of the discrepancy or problem, a defect number will be assigned to the defect. The defect will be numbered, logged, and entered into a defect repository. Unisys will review the defect repository frequently during testing, and weekly afterward, to determine its status and review it with NSP. For each test, defects will be assigned a classification as described in **Table 4.4-3** below.

**Table 4.4-3. Defect Severity Classifications.**

Incident Severity Code	Defect Severity Codes
C	<b>Critical:</b> A Critical Defect causes the software to be inoperable or materially malfunctioning in a way that renders it unusable. A Critical Defect does not have a reasonable detour or workaround and prevents a user completing normal operations or causes data corruption.

## MSS Test and Evaluation Master Plan (TEMP)

Incident Severity Code	Defect Severity Codes
S	<b>Significant:</b> A Significant Defect causes a serious malfunction, which is a deviation that would be considered a Critical Defect or impacts the business but has a reasonable detour or workaround. A reasonable detour or workaround is a process or procedure that allows the completion of normal operations without sacrificing data integrity and system reliability.
M	<b>Minor:</b> A Minor Defect affects but does not prohibit the user’s ability to complete their work. A “Minor” defect is a system defect that will be corrected in the future but does not materially prevent the end user from performing duties within the application. These types of issues would include cosmetic changes, a problem that is not evident to the user or other non-functional deficiency.
I	<b>Informational:</b> An informational defect is a documentation error or request for information.

### 4.5 Defect Correction and Verification

Unisys Team and NSP will review all defects before classification and after resolution. Each problem will be assigned to a specific member of the development team for resolution. Unisys will test the solution internally. Once the solution to the problem is tested internally, Unisys will perform the resolution, and NSP will verify it by executing the relevant portion of the test script. Relevant portions of test scripts can include the entire script or the specific step in the script for the functionality of the defect. Unisys Team will re-address the defect if the NSP team determines the defect has not been resolved.

### 4.6 Regression Testing

As defects are corrected and validated, there may be a need to validate that a correction did not have an unexpected impact on other system functions, processes, procedures, or components. This regression testing is performed by executing selected additional scripts or script sections. Additional test scripts to be executed in regression testing will be selected in accordance with the analysis of the defect and its associated resolution. Regression testing will apply to all test types and occur throughout the project.

### 4.7 Test Cycles

Testing will be executed in test cycles. The standard test cycle model is a three-cycle test. In this model, the first testing period cycle is started, and an attempt is made to execute all of the associated test scripts without stopping to resolve defects. This decision to stop will be balanced by the validity of completing a large portion of script testing instead of the number of defects encountered and what is



## MSS Test and Evaluation Master Plan (TEMP)

required to resolve particular defects. With this approach, the intent will be to use the first cycle of testing to identify all possible defects. The second cycle will include testing that occurs after defects from the first cycle are corrected. It will include the scripts that failed during the first cycle and scripts identified for regression testing and defect verification needs. The third cycle will represent a much more limited test cycle and focus solely on Critical and Significant defects that prevent system acceptance.

The order of execution of tests is shown in **Table 4.7-4**. This order will be followed after each release of software in the test environment.

**Table 4.7-4. Order of Testing.**

Order	Description
1	<b>Build Test</b> – This verifies that the application can be built and installed in the test environment successfully. The build/installation instructions included with the Release Notes document serve as the test procedure. The results are verified by the development team.
2	<b>Sanity/Smoke Tests</b> – A limited number of tests executed to verify that very basic functionality can still be performed immediately after each release of software.
3	<b>Defect Tests</b> – Test scripts associated with defects found in a previous release of software, which have been marked by development as ready for re-test.
4	<b>Test Cases</b> – The test procedures that have been created to test the functionality of the capabilities delivered in the increment.
5	<b>Regression Tests</b> – Test scripts that prove that the new release didn't break any previously tested features.

## 5. Entry and Exit Criteria

### 5.1 Test Plan

#### 5.1.1 Test Plan Entry Criteria

The entrance criteria specified below must be fulfilled before System, Factory Acceptance, System Integration, or User Acceptance Testing can begin:

- All developed and/or modified software has been software unit tested, factory acceptance tested, and software-integration tested by the Development Team.
- All Priority 1 and 2 defects identified in software unit/integration testing have been corrected.
- Test documentation, specifically the System Test Plan and Test Cases, have been reviewed and approved.
- Testing has received Release Notes from Configuration Management that include a list of which software objects have been changed or updated and which known defects are included or corrected.
- Test software is installed in the appropriate test environment and installation and sanity tests can be executed without defects.
- All test hardware and environments are in place and available for test use.
- All external dependencies identified have been satisfied.
- Personnel needed to perform testing are available and familiar with the application.

#### 5.1.2 Test Plan Exit Criteria

The exit criteria detailed below must be achieved before testing is considered complete:

- All test cases have been executed at least one time.
- There are no Critical Defect Severity Codes.
- Test Report is delivered to and approved by the NSP and Unisys Team Project Managers.