Medicaid Management Information System Replacement (MMISR) Project

MMIS Project Test Management Plan (PMO14)

HSD Deliverable Owner: Karin Stevenson
Deliverable Owner: EPMO
Configuration Number: V1.6
Date: 6/23/2022

Table of Contents

Table of Tables

Table of Figures

1.0 Introduction

The Test Management Plan (TMP) (PMO14) will serve as the framework and guide to testing activities for the New Mexico (NM) Human Services Department (HSD) Medicaid Management Information System Replacement (MMISR) Project for the Health and Human Services (HHS) 2020 enterprise solution in accordance with the Centers for Medicare and Medicaid Services (CMS) Testing Framework Overview. The TMP will detail the testing approach and methodology for use by module contractors and all project team members for their respective system testing. This TMP is based on and supersedes the previously approved PMO14. This TMP will follow the annual deliverable review process and will be updated at least annually. As the HHS2020 testing approach is more fully defined by the NM HSD Test Manager and/or as additional Module Contractors are onboarded to the project, the TMP may be subject to a more frequent update schedule.

1.1 Test Management Plan Purpose

This TMP describes the approach to managing and maintaining the MMISR Project testing lifecycle. The TMP outlines and communicates the intent of the testing effort for the MMISR Project incorporating CMS guidance on Certification and appropriate Security standards to define testing protocols. To support effective and efficient bi-directional traceability the MMISR project is leveraging the State's Jira and Xray tools as Test Management Tools (for more detail on requirements management, please review PMO15: Requirement Management Plan, which is listed in Appendix C. This plan focuses on the processes and stages for detailed testing methodologies to support the MMISR project needs, including traceability. Traceability is defined in detail within the Requirements Traceability Matrix (RTM) (PMO16). The use of tools is defined within the tool documentation found on SharePoint, with the links listed in Appendix D. This plan is not replacing HSD end to end integration test plan or any module contractor test plan. This plan is the overarching governance document that those test plans would fall under.

1.2 Test Management Plan Scope

The scope of this TMP is to provide an overall testing approach for the MMISR project to verify and validate the hardware and software infrastructure, interoperability, and enterprise business workflows of all the constituent systems, including modules, external and internal interfaces, and service orchestration.

All HHS2020 Module Contractors must comply with this TMP for integration into the enterprise. The EPMO Contractor, in collaboration with the Test Manager, will provide oversight, guidance and monitoring to Enterprise Stakeholders to ensure compliance with this plan. This plan will be subject to the annual deliverable review cycle and as additional Module Contractors and other agencies of the HHS2020 enterprise are onboarded, this TMP will be expanded to incorporate their testing.

The audience intended for the TMP is the MMISR project team, which includes, but is not limited to: Quality Assurance (QA) Testers, QA Test Managers, User Acceptance Test (UAT) Testers, UAT Test Manager, Module Owners, Contract Owners, Project Managers (PM), Application Developers, Infrastructure and Operations Support, System Security Managers, NM HSD's Information Technology Division (ITD), NM HSD's Medical Assistance Division (MAD), Certification, HSD Data Owners (Part of DGC), and Staff Augmentation team. Also, any other Stakeholders whose leadership and support are necessary to successfully implement this plan. Each Module Contractor will have testing deliverables included in their Statement of Work (SOW) and contract. The Module Contractor specific test plan is necessary to ensure that their solutions and software applications meet expectations. This TMP is to

ensure that those Module Contractor solutions can be integrated and functional for the MMISR and HHS2020 enterprise system. HSD, Module Contractors, Module Owners, and System Integrator (SI) are expected to be collaborative in testing efforts and support the UAT phase, end-to-end testing and production readiness testing required for both the MMISR system components and the agencies, and systems needed for the HHS2020 enterprise. To this end, the governing bodies for the HHS2020 will play a role in review and approval for confirmation of testing results and/or go/no go system go-live decisions. For a listing of all the project's governing bodies and their charters, please see Appendix C.

For clarity and use of this TMP, the acronym QA, or Quality Assurance, in this plan refers to the actual work performed by QA Testers, QA Test Manager, and/or the Quality Assurance work completed during testing phase; it does not refer to the Business Process Outsourcing (BPO) Module Contractor that is commonly referred to as the Quality Assurance Module Contractor on the MMISR Project.

2.0 Testing Framework

The testing framework overview is comprised of numerous testing functions that are conducted during the life cycle of systems and components in the enterprise solution. The project's overall testing approach and strategy is documented in this TMP along with detailed descriptions for each of the planned tests. Key detailed aspects of the specific testing approach, such as content, methodology, prioritization, and progression of testing activities will be documented in each Module Contractor's specific test plans for their offered solutions. NM HSD will develop a comprehensive testing plan that covers all functionality being integrated into the MMISR system. The comprehensive testing plan will be documented to include all the systems that will be integrated from the Module Contractors, and which will be tested.

2.1 Business Application Testing Overview

Business application testing involves four (4) categories of tests, as mandated by CMS, and are listed below:

- Development Testing
- Validation Testing
- Implementation Testing
- Operational Testing

CMS Testing Framework is included in this TMP in several places – 2.3 Testing Methodology, 3.0 Testing Process Flow and Appendix E. More detail to explain these four (4) categories of test can be found in that appendix. User acceptance testing is a component of Implementation Testing.

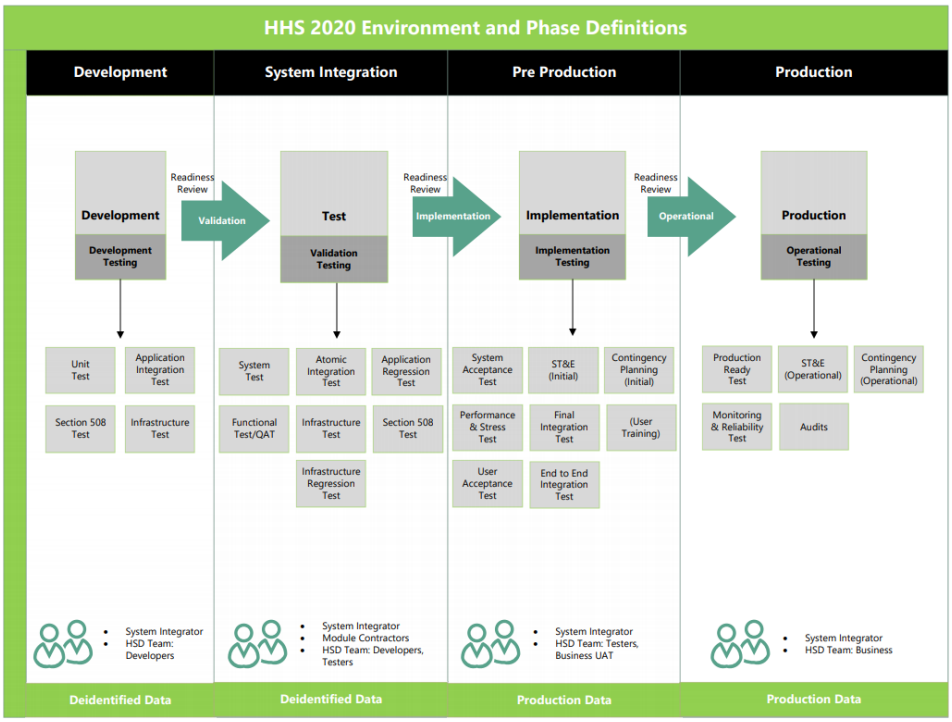
2.2 Infrastructure Testing Overview

The infrastructure-testing framework provides guidance on testing the infrastructure of systems involved in the MMISR Project solution as well as testing the software platform components that have been installed and configured. Installation and inspection of the infrastructure environments occurs for the test environments prior to the start of software testing. Infrastructure testing is also performed when a hardware or software platform goes through modifications to ensure that the changes have not caused unintended impacts, and that the system still complies with its requirements.

2.3 Testing Methodology

The overall testing methodology for the MMISR is guided by CMS' Expedited Life Cycle (XLC) Process which includes the development, test, implementation and operations and maintenance (O&M) phases. Within each of these phases, we have described a type of testing which will occur in each phase in the table below. Figure 1 shows CMS' Integrated IT Investment and System Life Cycle Framework; within this graphic, the 4 categories of testing phases are shown and below them the testing types.

Figure 1 - Environment and System Life Cycle Framework Framework



2.3.1 MMISR Testing Phases and Types

Every Module Contractor team's individual TMP must identify the applicable testing practices from the testing phases identified in Table 1 that are required to effectively test the functionality required of their solution.

The following tables have been updated to include the CMS test phase, the MMISR testing phase, a description of the testing phase, type of data used in the test phase, and whether the test phase occurs in a module contractor. The environments listed have been aligned to the environments that HSD has

planned to build for the MMISR project. Where UAT is called out as a testing phase – User Acceptance Testing is meant as the testing phase and not an environment.

Table 1 - MMISR Testing Phases

CMS Category of Testing	MMISR Testing Phase	Description	Type of Data Used for Testing Phase	Module Contractors (MC) Environment	HSD Environments	MC Responsibilities for testing within the HSD environment	SI Responsibilities for testing within the HSD environment
Development	Unit Testing	Unit testing is performed by the development team after, or in parallel with, application development to assess and correct the deficiencies of individual software units	Non-production data If available, and if permissible, production data will be used as test or mock data does not replicate real-life scenarios	Development - MC	Development - HSD (for SI use only)	Module Contractors will perform unit testing for their software solutions in own environments	SI will perform unit testing in the HSD environment for shared services solutions in HSD environments; SI will also perform unit testing of interfaces and will use HSD environments
Test	Integration Testing	Integration testing focuses on integrating an individual software system with one or more internal/external systems/modules/components/services/interfaces. In this phase, the module contractors come together to test the fully integrated solution required to achieve a successful business workflow ****Note**: The scope of testing that is required for COTS software*	Non-production data If available, and if permissible, production data will be used as test or mock data does not replicate real-life scenarios	Testing environments - MC	System Integration Testing (SIT) - HSD	Module Contractors will perform / participate in integration testing activities for their software solutions and services after unit testing in their own environment is completed	SI will coordinate integration testing activities to ensure that module contractors are not encountering testing conflicts when they are performing integration tests in the HSD SIT environment. Further, SI will coordinate the interactions between module contractors for the testing of an integrated solution or solution component

CMS Category of Testing	MMISR Testing Phase	Description	Type of Data Used for Testing Phase	Module Contractors (MC) Environment	HSD Environments	MC Responsibilities for testing within the HSD environment	SI Responsibilities for testing within the HSD environment
		products may be reduced based upon an individual assessment or situational assessment of the COTS product.					
Test	System Testing	System testing is limited to end-to end functional validation with real interfaces using predefined system test cases and test data. The end-to-end system testing checks the HHS 2020 enterprise system's ability to maintain data integrity and operation in coordination with other systems. Section 508 Accessibility verifications are most often performed during the System Testing phase	Non-production data If available, and if permissible, production data will be used as test or mock data does not replicate real-life scenarios	Testing environments - MC	SIT Limitations could include: • SIT testing will not be at an Enterprise level but will require atomic integrations, with connections to endpoints outside of our environment's ecosystem (to FS, DS, BMS, etc.) • No load testing in SIT, no E2E process testing, no E2E orchestrations. J ust 'bits and pieces'. • No E2E or performance testing can be done in SIT due to expected lack	Module Contractors will perform/participate in system testing activities for the functional validation of their software solutions within the enterprise system	SI will coordinate the system testing phase and conduct the end-to-end functional validation of system test cases

CMS Category of Testing	MMISR Testing Phase	Description	Type of Data Used for Testing Phase	Module Contractors (MC) Environment	HSD Environments	MC Responsibilities for testing within the HSD environment	SI Responsibilities for testing within the HSD environment
					of availability of E2E data sets.		
Test	User Acceptance Testing (UAT)	UAT is performed to verify the overall MMISR enterprise solution as per the acceptance test cases. This testing takes place in the final phase before moving the software application to the production environment. This test phase is where the customer validates, whether the solution meets the business requirements. Section 508 Accessibility testing for selective scenarios as requested by business owners can be performed during the UAT test phase by the UAT test team.	Production data	N/A	– Pre-Prod	HSD Conduct and/or coordinate testing activities with MCs for the solutions and how they integrate with MMISR	SI will be available to participate and/or support HSD for UAT testing.
Implementation	Production Readiness Testing	Production readiness testing is part of the operational readiness evaluation and is performed to confirm that a production-ready application and infrastructure have been installed and configured	Production data	N/A	Production - HSD	MAD Business Operations Team (and may coordinate with module contractor testers) will conduct production readiness testing and verify that functions meet business expectations	SI will be available to participate and/or support HSD and the MAD Business Operations team for Production Readiness testing

CMS Category of Testing	MMISR Testing Phase	Description	Type of Data Used for Testing Phase	Module Contractors (MC) Environment	HSD Environments	MC Responsibilities for testing within the HSD environment	SI Responsibilities for testing within the HSD environment
		correctly in the production environment and is ready for operational use. The business operations team may choose to use backlogged work to support production readiness testing as this will replicate real productions work as close as possible.				and functionality to run operations	

NM HSD expects all Module Contractors to perform all phases of testing for their offered solutions.

NM HSD has retained the UAT phase of the MMISR project and will be performing UAT in-house on individual modules from each vendor and on the integrated MMISR system as needed. UAT is a critical phase of the Design, Development, and Implementation (DDI) life cycle as it is one (1) of the last opportunities to verify that the system is working as expected, prior to migration of a new module or system into production.

The Table below outlines the testing types and provides a description of each type. The environment column designates which environment the testing will occur in. Functional testing may be conducted in the Production environment based on the need. The Responsible column designates the responsible party that will conduct the type of testing.

Table 2 – MMISR Testing Types

Testing Types	Description	HSD Environment	MC Responsibilities for testing within the HSD environment	SI Responsibilities for testing within the HSD environment
Infrastructure Testing	Infrastructure testing focuses on the hardware and software platform on which the MMISR solution systems run on. The	DEV, SIT, Pre-Prod and Production	Module Contractor who provisions the infrastructure that supports the module contractor's software solution will test their own infrastructure	The SI will test the HSD infrastructure if they provision the servers and environments

Testing Types	Description	HSD Environment	MC Responsibilities for testing within the HSD environment	SI Responsibilities for testing within the HSD environment
	objectives of infrastructure testing include platform provisioning, network connectivity and access, security testing of infrastructure, software platform installation and configuration, as well as a readiness check for application use			
Integration Testing	Integration testing focuses mainly on integration testing of the individual software unit with one or more internal/external systems/modules/ components/services/ interfaces. In the absence of a real partner for integration, mock services and mock data shall be used to perform unit integration	SIT and Pre-Prod	Module Contractors will perform/participate in integration testing activities for their software solutions and services after unit testing in their own environment is completed.	SI will coordinate integration testing activities to ensure that module contractors are not encountering testing conflicts when they are performing integration tests in the HSD SIT environment. Further, SI will coordinate the interactions between module contractors for the testing of an integrated solution or solution component
Breadth Verification Testing (BVT)/Smoke Testing	BVT/Smoke Testing is the first step in testing after each build is deployed on the test environment and it verifies that the essential and core functionality of what is delivered works and that the build is acceptable for further testing efforts. BVT/Smoke Testing emphasizes breadth instead of depth. BVT/Smoke Testing failure leads to rejection of a build for use in further testing efforts	DEV, SIT and Pre-Prod	Module Contractor will perform/participate in BVT/Smoke testing activities for their software solutions and services.	SI will coordinate BVT/Smoke testing activities to ensure that module contractors are not encountering testing conflicts when they are performing tests in the HSD environments. Further, SI will coordinate the interactions between module contractors for the testing of the integrated

Testing Types	Description	HSD Environment	MC Responsibilities for testing within the HSD environment	SI Responsibilities for testing within the HSD environment
				solution or solution components
Module Contractor Regression Testing	Regression testing is performed on a build to ensure the recent changes have not adversely affected preexisting and validated functionalities or introduced new behaviors that are undesirable. In addition, the regression testing verifies that the newly introduced functionality and/or fixes work as intended.	Development, Integration, and Pre-Prod	Module Contractor that owns the change will test to confirm that new functionality works as intended and that no other functionality is adversely affected Module Contractor that is affected by the change will test to confirm their solution and system functionality is not impacted	
Functional Testing	Functional testing is the process of validating software to ensure that it conforms to functional, and user requirements. It is used to test the features/functionality of the system or module and covers all the scenarios including failure paths and boundary cases. Mock services and mock data are used to perform functional testing in absence of real system integration, if available. Functional testing also covers the usability and browser compatibility of web interfaces	SIT	Module Contractors will perform/participate in Functional testing activities for their software solutions and services that are a part of the integrated solution	SI will coordinate Functional testing activities to ensure that module contractors are not encountering testing conflicts when they are performing tests in the HSD environments and to ensure that all software conforms to requirements
Performance Testing (non-functional)	Performance testing assesses the capacity and throughput of the enterprise application and/or infrastructure in terms of processing and response time, Central Processing Unit (CPU) utilization, network utilization, and memory and storage capacities	SIT and Pre-Prod	Module Contractors will perform/participate in performance testing activities for their software solutions and services that are a part of the integrated solution	SI will coordinate the performance testing activities of the integrated solution.

Testing Types	Description	HSD Environment	MC Responsibilities for testing within the HSD environment	SI Responsibilities for testing within the HSD environment
	relative to expected normal (average and peak) user and processing load. Performance testing is both how the system will perform when hit normal and to show when it will break (stress/load testing). Stress/load testing mainly measures the system on robustness, capacity, and capabilities under extremely heavy load conditions. It can be used to determine the system breaking point beyond normal operational capacity to observe the results			
Security Testing (non-functional)	Security testing is a variant of software testing which strives to detect that systems and applications are free from any security vulnerabilities and validate that security controls (technical and management) are implemented according to the Security Design Plan (SDP) and System Security Plan (SSP). Security testing also strives to detect that unauthorized user access to confidential data is prevented. Last, it involves checking that the correct users or user types are able to access systems appropriately	Dev, SIT, Pre-Prod, and Production	SIT – Module Contractors, Pre-Prod – HSD, and Independent	SI will coordinate the security testing activities of the integrated solution with HSD and module contractors
Parallel Testing	Parallel Testing in the HHS2020 context is a testing type in which	Pre-Prod	Module Contractors will perform testing of key system functionality and	SI will coordinate and/or oversee Parallel Testing

Testing Types	Description	HSD Environment	MC Responsibilities for testing within the HSD environment	SI Responsibilities for testing within the HSD environment
	multiple versions of equivalent processes or applications are tested with the same input. The outputs and results are compared to ensure that they are equivalent as well.		complete business process testing of a subset of work while at same time HSD UAT and MAD testers test business processes to compare results.	with the module contractors and HSD & MAD resources for the offered solutions and for the integrated system.
Section 508 Testing	The objective of Section 508 compliance testing is to ensure the user interface and any output generated from the application is compliant with applicable Section 508 Accessibility Standards identified in the Section 508 Product Assessment (considered part of Functional Testing)	Dev, SIT, and Pre-Prod – except Production	Module Contractors will perform 508 testing for their offered solutions and system components	Coordination will occur between the appropriate parties to conduct 508 testing with module contractors for their offered solutions and for the Unified Portal
User Acceptance Testing (UAT)	UAT is performed to verify the overall MMISR enterprise solution as per the acceptance test cases. This testing takes place in the final phase before moving the software application to the production environment. Acceptance testing is performed by a business owner to validate the business requirements are met and are facilitated by an HSD testing team. Selective functional, integration, and system test cases may be utilized. In addition, a "mini-regression" test may also be conducted at the start of each UAT iteration to ensure there	Pre-Prod	N/A	SI will be available to participate and/or support HSD with UAT, as necessary or required

Testing Types	Description	HSD Environment	MC Responsibilities for testing within the HSD environment	SI Responsibilities for testing within the HSD environment
	are no adverse effects for verified functionalities			
End-to-End Testing	The End-to-End testing phase is planned upon completion of the individual module specific tests and is conducted to ensure that the system behaves cohesively and as expected	Pre-Prod	Module Contractors will participate and/or support HSD with end-to-end testing, as necessary or required	SI will be available to participate and/or support HSD with End-to-End testing, as necessary or required
Contingency Plan Testing / Disaster Recovery (DR)	The objective of contingency plan testing is to verify the success of the restoration procedures that are executed after a critical IT failure or when disruption occurs. The Disaster Recovery Plan (DRP) protects resources, minimizes customer inconvenience, and identifies key staff. It also assigns specific responsibilities in the context of the recovery	DR Environment (as available), Module Contractor Environments	Module Contractors will participate in contingency plan testing and DR testing with HSD and other module contractors	SI will coordinate and lead contingency plan testing and DR testing with HSD and module contractors' participation

2.4 Testing Deliverables for Module Contractors

Testing deliverables are comprised of system test plans, system test reports, test cases, test results with evidence of execution, and system acceptance test plans (e.g., Data Services (DS) – data warehouse) that are provided by the Module Contractor testing teams. These deliverables are used to update the RTM. Additional information can be found in the Requirements Management Plan (PMO15).

NM HSD has a review, advise, and approval role for all Module Contractors testing deliverables during their DDI phases of their solution development. Descriptions of the types of test deliverables that NM HSD will review and approve are listed below. Module contractors will not have specific UAT deliverables as NM HSD is responsible for UAT testing including test plans, test cases, and test execution. Module Contractors will be required to provide documentation and information for entry criteria for UAT testing to NM HSD for their offered solutions.

2.4.1 Test Plan

A test plan is derived from functional and non-functional requirements and detailed design specifications. The test plan identifies the details of the test approach, identifying the associated test case areas within the specific product for the release cycle. The test plan serves as a blueprint to conduct software-testing activities as a defined process.

2.4.2 Test Cases

Test cases are a set of pre-conditions or variables that provide detailed steps under which a tester determines whether an application or software system is working as expected. Test cases are based on requirements and documented in the Xray for Jira tool for easy maintenance, traceability, and organization. Test case status is tracked in the RTM.

2.4.3 Test Sets (Optional)

Test sets are a group of test cases that belong to a feature, or which can be executed together. A test set contains detailed instructions for each collection of test cases, as well as information on the system configuration to be used during testing. Test sets are identified while creating test plans and documented in the Xray for Jira test tool. Test sets are optional for Module Contractors to use, as the priority for test execution is ensuring adequate test plans and test cases are created that can be replicated and repeated, as necessary.

2.4.4 Test Execution

The test execution is the activity that occurs when test cases are run once the test target is in an appropriate state to have tests run against it. Test execution for each test case needs to contain evidence of execution. Test execution evidence can be manual or automated, depending upon tools used by the Module Contractor. Test results only pertain to a specific product version and environment. Module Contractor testing tools must be approved by the State and integrate and support the State's approved RTM tools and processes. Test Execution is an issue type within the Jira system.

2.4.5 Test Reports

The test report is a document that contains the test results and the summary of test execution activities. The test report consists of:

- Number of test cases created
- Number of test cases executed
- Percentage of test cases that were passed, failed, blocked, or not executed
- Defect reports with a summary of total defects opened, closed, and deferred during the testing process. Also contains defect status and severity
- Monthly status report containing the execution and progress of testing activities by testing teams

In addition, customized test reports are generated for different stakeholders depending on the information requested.

2.5 Test Readiness Reviews

Test Readiness Review (TRR) meetings are conducted to determine if the MMISR Project system under test is ready to proceed from one (1) testing phase to the next.

The TRR assesses the test objectives, test methods and procedures, and the scope of tests. It confirms that test results from the completed test phase are complete and accurate. The TRR also verifies that the test cases, environment, test data, and test resources have been prepared for the next test phase. The TRR verifies the traceability of planned tests to the requirements and test plan.

2.5.1 Test Readiness Roles and Responsibilities

The following table lists the TRR actors, along with their roles and responsibilities. The roles include all the Module Contractors participating in the testing activity. TRR will be a coordinated effort between SI and the other Module Contractors which is initiated upon the readiness of the module contractor to begin.

Table 3 - Test Readiness Roles and Responsibilities

Role	Responsibilities
HSD UAT Test Manager	▪ Work with Module Contractor Test Lead ▪ Review Module Contractor test plans, test cases, and testing deliverables ▪ Observe and review test cases for execution on Module Contractor's offered solutions ▪ Review results of approved test reports from Module Contractors ▪ Review and approved evidence of execution of testing by Module Contractors
HSD Requirements Manager	▪ Collaborate with the HSD Test Manager and the Module Contractor Test Leads to support requirement traceability
Module Contractor Test Lead	▪ Presents the evidence of test execution in the form of test reports and test cases from all testing phases ▪ Provides the test cases, data, and environment identified for testing phases ▪ Communicate with HSD regarding Module Contractor's solutions, testing efforts, and integration testing purposes ▪ Write test cases for execution

Role	Responsibilities
Lead Developer	▪ Presents the evidence of unit test execution, code coverage, and quality reports from the development testing phase
Project Manager	▪ Tracks the testing activities against the project schedule ▪ Oversees the testing readiness review process
Certification Manager	▪ Verify the evidence mapped by the Module Contractors for the certification process for their solutions
Security Lead	▪ Oversee the security portions of all testing efforts ▪ Review and develop test cases for security ▪ Participate in TRR meetings
IV&V and Other Stakeholders	▪ Reviews the testing evidence of the enterprise application by inspecting test reports of all test phases and preparedness of each test phase testing
Change Control Management Plan (CCMP) and Change Control Boards (CCB)	▪ The change boards within the CCMP approve/disapprove TTR artifacts in accordance with CMS policies, plans, guidance, processes, and procedures

2.5.2 Process Activity for Test Readiness Review

Table 4 shown below is included here as an example of information needed to conduct a TRR meeting. For all solutions offered by Module Contractors, a process activity documented like the Table 4 will be used to determine if the MMISR Project system under test is ready to proceed from one (1) testing phase to the next. The following are examples of the inputs and outputs, entry and exit criteria, and the process activities involved in conducting a TRR. TRRs will also be used to define entry and exit criteria for integration testing between Module Contractors. Each Module Contractors and NM HSD testing team will establish these expectations relative to the specific functionality being tested and test readiness reviews will be approved by the NM HSD Testing Manager and agreement about testing process and outcomes will be gained before testing phase begins. In the table below in the process activity section, TRR1 is meant to indicate a single, standalone Module Contractor solution being tested during TRR1. TRR2 is for integration testing and will be performed before proceeding to UAT. Coordination with the Module Contractors to complete the TRR1 and TRR2 processes will then have to be aligned with business users to move into UAT Phase.

Table 4 – Example of Test Readiness Process Activity for TRR

Entry Criteria	Input
▪ Test case execution of the previous testing phase is completed with no outstanding critical and major defects ▪ Test reports are documented according to Appendix E ▪ Test cases, test data, and environment are identified for the next testing phase	▪ Test case and execution test reports from previous testing phases ▪ Test cases and test data identified for the next testing phases
Process Activity	
All participating Module Contractors will provide the evidence of the testing activities, including the test reports and metrics, and confirm the successful completion of testing activities of the previous phase before proceeding to the next testing phase.	

The review board actors, listed in Appendix E, shall then review the evidence, inspect the quality of the testing activities, and qualify the release to the next testing phase.

In the scope of the HHS2020 enterprise testing strategy, each system that undergoes testing phases should go through at least two (2) TRRs. The following are the details pertaining to the timing of the reviews that will be performed:

- **TRR1:** TRR1 shall be performed before proceeding to system testing. The following are the artifacts that shall be presented as evidence for this review process:
  - Unit test case report from development testing
  - Test cases documentation and execution report from QA validation testing including the following:
    - Functional test cases and execution report
    - Integration test cases and execution reports
  - Section 508 test cases and execution reports, if applicable
  - Pre-requisites for testing in all lower environments must be met before moving on to TRR2

- **TRR2:** TRR2 shall be performed before proceeding to the UAT phase. The following are the artifacts that shall be presented as evidence for this review process:
  - Reports from the previous TRR
  - Performance and security test cases and reports
  - Infrastructure and contingency plan (disaster recovery) test reports

Exit Criteria	Output
▪ Test cases, plans, and execution reports from previous phase are reviewed and approved ▪ Mitigation strategy for open defects is established ▪ Test cases, and test plans, for the next phase are reviewed ▪ Environment for the next phase is ready for performing testing activities	▪ If the result of the review process is satisfactory with no outstanding critical and major defects and, if necessary, if a viable workaround is identified for a defect, the release shall be qualified for the next testing phase only on an exception basis where the major defect can be made a medium based upon viable workaround. Defect will be left open for resolution while continuing onto the next phase ▪ If not satisfactory, the release will be rejected to the previous phase

2.6 Test Data and Simulation

Test (mock) data is the input data created in accordance with and used during the execution of test cases. Though test data can be generated manually by a tester or a test team testing a Module Contractors' solution, as part of test case preparation, there are some disadvantages in using manual test data, such as:

- Manually generated test data requires data to be prepared every time a test case is executed
- Manual generation of test data for all possible scenarios requires immense time and effort
- In some cases, and with NM HSD approval on a case-by-case basis, if permissible, and available, production data will be used to ensure adequate testing effort can be achieved as mock data often does not simulate any real-life issues present in production data from legacy systems

Automated test data, though it requires upfront effort, can help generate repeatable test data covering the wide possibility of ranges, saving time and effort in the long run; however, test cases and use of test

data needs to be created with exceptional attention to detail to ensure it will realize and simulate as many real-life issues as possible.

In the case of the HHS2020 enterprise solution, test data is consumed by the following categories of application:

- **Data-driven Applications**
  - Test data from the equivalent environment of the integrating systems are copied to the target test environment. For example, data from the SIT environment of the integrating systems to the SIT environment of the new module. For example, test data for the provider enrollment solution (from Benefit Management Services module contractor) may need to be moved to the integration environment so that Financial Services module contractor can utilize this data to conduct testing for pending claims.
  - The mock data is generated from the production environment, where data is exported from these environments and sent through a masking program, which removes or replaces sensitive Personally Identifiable Information (PII)/Protected Health Information (PHI)/ from the source data and substitutes it with random data
- **Service-driven Applications**
  - Accessing test data using services published by the test environments of integrating system
  - A mock service is created when there are any external or internal systems currently in development that do not have fully developed Application Program Interfaces (API) and web services. To enable parallel development, mock services are utilized to mimic the exact behavior of the actual APIs/web services hosted by these modules
  - The mock service can also be utilized when there is no matching testing environment hosted by either external systems or internal /systems hosted by the module contractors corresponding to the test environment of any HHS2020 system

2.6.1 Test Data Roles and Responsibilities

The MAD business resources will have to be partnered with the testing personnel to ensure validation of mock data is completed in a manner that the business can verify is logical and supportive of business use cases. The following table lists the actors in preparation and utilization of test data:

Table 5 - Test Data Roles and Responsibilities

Role	Responsibilities
Software Developer	▪ Creates test data for unit testing ▪ Develops mock services of services provided and required to facilitate integration testing
Tester	▪ Uses test data to perform functional testing ▪ Validates mock data and services for correctness and completeness of the business function with MAD business resources ▪ Uses mock services to simulate integration between modules and confirms success of testing with MAD business resources

2.7 Test Data Design Methodology

The objective of the test data design methodology is to leverage the features offered by the database and middleware systems to provide mock data and services. If the existing systems do not offer such

features, the testing team will make use of the Commercial Off the Shelf (COTS) testing products that offer mocking mechanisms. These testing products will be offered by the SI and/or other Module Contractors when they propose their solutions.

The two (2) types of designs are:

- **Mock Data Design** - Mock data is generated from production or production-like systems to enable testers to execute their test cases with near real-time data to ensure that the software functions effectively in the production environment
- **Mock Service Design** - Mock services are generated for external systems that do not have a testing/sandbox environment matching the MMISR systems or in the case where the APIs which handles interactions between software components and passes data back and forth are not developed yet or the BPO module contractor is not onboard the project yet.

2.7.1 Test Data Disposal

Test data disposal (tear down) is the process of cleaning test data used during test case execution to enable its reuse. Tearing test data down is an integral part of the test suite execution. Tear down scripts are required for the following reasons:

- Test data once utilized by a test case on an environment cannot be reused by the same test case without cleaning the data from the database
- Over time, test data accumulates in the databases of environments and can cause erratic behavior during test case execution. To maintain databases in a pristine condition, an effective tear down mechanism needs to be in place

SI and module contractors will propose the best test data disposal methods through their Test Plan deliverables which must be approved by HSD.

2.8 User Acceptance Testing Approach by NM HSD

According to CMS Testing Framework, "**User Acceptance Testing (UAT)** is an application testing performed by a Business Owner to validate that the business requirements are met, which may be facilitated by a Testing Contractor."

The HSD UAT team is adopting a modular approach to address the UAT needs of the MMISR project. The Module Contractors will be testing in isolation their own offered solutions; during integration testing, they will participate in testing efforts of their solutions and components that integrate with other Module Contractor's systems.

UAT Test Plans will be drafted for each module clearly identifying the scope, approach, methodology, and management along with module specific test bed and artifacts.

The modular UAT Test Plan and approach closely align to the CMS Testing Framework and HHS2020 Testing Phases and aligns the business requirements of the module from a business user perspective.

Below in Table 6, we have listed which Module Contractors are currently agreed to be in scope and out of scope for the UAT.

Table 6 - In/Out of Scope

In Scope	Out of Scope
▪ Benefit Management Services	▪ Other agencies participating in HHS2020*

In Scope	Out of Scope
▪ Provider Management Services ▪ Data Services ▪ Financial Services ▪ Pharmacy Benefit Management ▪ Quality Assurance ▪ Unified Portal (Internal and External Portal)	▪ Benefit Management Services ▪ Care/Case Management Services**

** Anything other than MMISR modules is out of scope for UAT*

***Care/Case Management Services will be delayed implementation and agency ownership of this module may transfer from HSD to another agency in the future*

NM HSD has expectations that Module Contractors will complete prior phases of testing as planned for each of the modules in scope by respective vendors before NM HSD will begin their UAT, as required and outlined in Section 2.5 Test Readiness Review. Adherence to scope, schedule, and quality delivery by module contractors is key for successful initiation, progress, and completion of UAT testing both at module level as well as end-to-end phase. The SI will participate in UAT; supporting the shared services for the modules that are using them. In addition, each module and its associated subsystems will be included in the module contractor's specific test plans. This TMP is not meant to convey the level of specificity or detail of each module contractor's test management plan.

Additional details, on a more granular level for UAT testing, will be established with each Module Contractor and will evolve over time as the modules complete their DDI phases.

2.9 Section 508 Accessibility Testing

Accessibility Testing (aka Section 508 testing) serves to validate that applicable Section 508 Accessibility Standards for user interfaces and any electronic output are met. As of February 2021, WCAG 2.0 and WCAG 2.1 (for mobile interfaces) are the foundation for Accessibility verification.

The CMS policy as specified in https://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/Section508/Section_508_policies_procedures, including the CMS Section 508 Policy document available on the CMS Section 508 Policy website, are applicable for HHS2020 software deliverables. In the following quotes from CMS policy sections, the term ICT stands for "Information and Communication Technology".

The policy's scope states "This policy applies to all CMS employees, contractors, interns, and other non-government employees performing CMS business and all partners (***including organizations receiving grant funds***) or organizations collecting or maintaining information or using or operating information systems on behalf of CMS (herein referred to as the CMS Community)." The policy further states "It is the policy of CMS that ICT developed, procured, maintained, ***funded*** and used by the agency will be accessible to persons with disabilities". Any exceptions and measures to gain exceptions are detailed in the exception type section of the CMS Section 508 Policy document.

Module Contractors are encouraged to complete or provide an existing Section 508 Voluntary Product Accessibility Template (VPAT). For details and access to the template please refer to https://www.section508.gov/sell/vpat.

Section 508 testing applies to the Unified Portal (UP) and any of the module contractor's BPO-offered solutions. The SI platform itself is subject to limited Section 508 testing as applicable for system administrators.

Section 508 testing benefits from the use of testing tools that evaluate user interfaces for compliance with the WCAG standards. That includes screen reader tools, color analyzer tools, code structure analyzer tools, etc. Examples of the most widely used tools include tools WAVE Analyzer, ANDI, NVDA screen reader, and JAWS screen reader (requires license purchase). However, certain situations will require manual testing by human testers to assess the compliance, e.g., contrast assessments for font color/size in the foreground of images.

Most tools do not provide "certification" of 508 compliance. They are tools that testers can utilize for assessing the compliance of each user interface element as part of documented test scenarios. HSD reserves the right to review the executed test scenarios and test execution evidence for Section 508 Compliance verification as part of contractually agreed upon test deliverables.

Note that the above-mentioned tool examples and the links in the Helpful Links section below do not constitute endorsements for the listed tools, nor are they a comprehensive list. Module Contractors should select their test methods and tools that accomplish the task for verifying complicity with the WCAG standards and CMS policy specifications as applicable.

2.1.1 Helpful links

Intro to Accessibility and 508:
https://www.hhs.gov/sites/default/files/Intro%20to%20Accessibility%20and%20508.pdf

Color and Contrast Requirements: https://webaim.org/articles/contrast/

Tips for Manual Contrast Assessments (from ANDI but generally applicable):
https://www.ssa.gov/accessibility/andi/help/alerts.html?manual_contrast_test_bgimage

Link to ANDI tool, installation, and tutorials: https://www.ssa.gov/accessibility/andi/help/install.html

Link to WAVE tool, installation, and tutorial: https://wave.webaim.org/

Link to NVDA tool, installation, and tutorials: https://www.nvaccess.org/

Link to WCAG 2.0 Standard: https://www.w3.org/TR/WCAG20/

Link to WCAG 2.1 Standard: https://www.w3.org/TR/WCAG21/

3.0 Testing Process Flow

As explained in Section 2.3 Testing Methodology, the overall testing methodology for the MMISR is guided by CMS' Expedited Life Cycle (XLC) Process which includes the development, test, implementation and operations and maintenance (O&M) phases. Further, the MMISR Project is at a Complexity Level 3, according to CMS' rating guidance on project risk. The figure below graphically represents the CMS XLC Phases of Planning, Requirements Analysis & Design, Development & Test, and Implementation.

A summary link to CMS XLC Phases is included in the Appendix C.

Commented [BM1]: Does any of this need updating based on the switch to TLC from XLC? I admittedly know very little on this subject

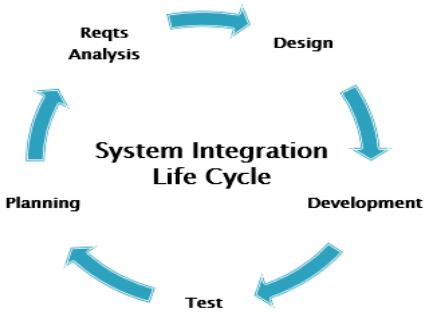
Commented [KPS2R1]: I don't know what this refers to.Yes

Commented [RO3R1]: Melvin and Liz aren't familiar with this and we have decided to keep that document as is for now. Liz will take on the task of researching this and we can always update this later.

Commented [RO4R1]:

Commented [KPS5R1]: https://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/TLC refers to the new CMS process. This document is still based on the XLC so it needs to stay as is.

Figure 2 – SILC



- **Requirement Analysis:** Requirements are key inputs to the testing process and the analysis of requirements is the first step towards designing a testing strategy
- **Test Planning:** During the test planning phase, the testing team will create test plan documentation and perform test tool selection, effort estimation, resource allocation, and training for testing activities
- **Test Case Documentation:** In this phase, the testing team will document the detailed test cases and prepare the test data needed for testing. Test cases and test data go through peer review by the appropriate testing team to make sure that the test cases requirements are met
- **Test Case Execution:** In this phase, the testing team executes test cases according to the test plan. Failed test cases are logged in Jira for tracking and reported to development for remediation. Re-running of selective test cases are performed, as applicable. Automated testing tools may be planned for use by a Module Contractor. If this is the case, NM HSD will expect the Module Contractor to provide an overview and demonstration of the specific automated testing tool prior to use for test case execution
- **Test Results Analysis:** In this step, the results of the test case execution are analyzed for continuous process improvement
- **Test Cycle Closure:** In this step, the testing team will evaluate the test completion criteria based on the test coverage tracked in the Xray tool. Once the test cycle is completed, a test closure report and test metrics are prepared. This also involves evaluation of the lessons learned for quality improvement of any further, or the next, testing cycles. The entry criteria for test closure are the thoroughly analyzed test results of all testing phases with no outstanding critical or major defects, or without sufficient viable workarounds for the defects that would allow them to be downgraded to a medium.

3.1 Defect Management

Defect Management is the process to track and appropriately support the timely resolution of defects identified during the testing phases of the MMISR Project. The processes outlined apply during the Design, Development, and Implementation (DDI) phase of the MMISR project up to go-live. Upon go-live, contractual obligations and Maintenance and Operation (M&O) defect resolutions processes will

apply. The Defect Management process is being updated for the MMISR project. A link to the current process is included in Appendix C and references to the final update will be at the same location.

4.0 Roles and Responsibilities

The EPMO, in collaboration with the HSD Test Manager, is responsible for oversight for the execution of the TMP. The MMISR Project Leadership team is responsible for successfully delivering the MMISR project with support from the Testing Manager (and various project teams/members). The EPMO provides guidance and direction to the Test Management process and acts as an escalation point to support project progress. As workstreams are initiated, individuals are assigned to the roles listed in the table below in conjunction with the HSD Test Manager and the Module Contractor PM.

Throughout all phases of the Project, the EPMO, IT Project Managers, the Module Project Managers, the Module Contractor Business Analyst (BA) Teams, and HSD Test Team will collaborate on testing efforts including planning, scheduling, stakeholder engagement, review, and approvals in order to ensure an integrated and consistent Test Management approach across the program is achieved.

Testing efforts for MMISR are led by the Module Contractor's Test Manager or Lead, who is focused on guiding the MMISR testing process for their module's offered solution and ensuring the successful delivery of the fully functioning scope of the MMISR project. The key areas of responsibilities are:

- Maintain the MMISR Test Management lifecycle
- Act as the point person for HSD for test management of module's offered solution
- Ensure that there is business representation within the testing process at appropriate testing phases
- Collaborate with the EPMO, IT Project Managers, BAs, and the Module Project Managers in ensuring adherence to this plan and supporting any updates
- Collaborate with the MMISR Requirements Manager to support requirement traceability
- Collaborate with the MMISR Security Manager to support the security requirement needs of the project
- Collaborate with the MMISR Certification Manager to support the certification testing needs of the project

NM HSD will lead testing efforts for the UAT phase; similar responsibilities apply to UAT testing phase. The following table outlines how MMISR resources have a responsibility for participating in, and following, the Test Management processes:

Table 7 - Roles and Responsibilities

Role	Responsibility
Module Contractor BA	Develops the requirements and manages them using the Jama tool. The BA is the liaison between the testing and development team and all project stakeholders in terms of requirements analysis and clarifications
Module Contractor Infrastructure Engineer	Responsible for infrastructure capacity planning and setting up the infrastructure
Module Contractor Database Administrator	Responsible for database setup and performing administrative tasks, such as troubleshooting, performance tuning, security audit, backup, and data recovery

Role	Responsibility
Module Contractor Solution Architect	Responsible for devising technical solutions to address business requirements. Defines the structure, characteristics, behaviour, and other aspects of software and presents them to project stakeholders. Provides specifications according to which the solution is defined, developed, managed, and delivered
Module Contractor Software Developer	Participates in requirements analysis, design, software development, unit testing, implementation, and maintenance support
Module Contractor Project Manager (PM)	Plans and coordinates the testing activities within the overall project schedule, tracks the testing activities, mitigates risks, and participates in testing readiness reviews (review gates) to ensure that the appropriate artefacts are generated for certification purposes
Module Contractor QA Testers	The following are the responsibilities of a Quality Assurance Tester: ▪ Creates, designs, executes the test plans and cases, and identifies test sets ▪ Develops and maintains testing standards and procedures ▪ Executes the test cases and test sets to validate the developed functionality in each testing phase ▪ Works with the development teams to resolve any defects that arise out of the testing process ▪ Participate in peer test case reviews
Module Contractor QA Manager	Responsibilities of QA manager are: ▪ Works with development leads and solution architects to confirm that testing tools, environments readiness, and security standards are in place ▪ Reviews the testing process to ensure that defects (identification, fixing, and re-testing) are addressed, and that testing standards, guidelines, and methodology are followed ▪ Participates in test case and test data reviews ▪ Tracks the requirements against the test cases and results for RTM
Module Contractor System Security Manager	▪ Responsible for the Delivery Security Design Plan (SDP) and System Security Plan (SSP) according to CMS requirements
HSD Certification Manager	▪ Responsible for coordinating with module contractors to assure evidence meets certification requirements
Data Steward	▪ Responsible for ensuring the quality and fitness for purpose of the organization's data assets, including the metadata for those data assets. Will share some responsibilities with a data custodian, such as the awareness, accessibility, release, appropriate use, security and management of data
Independent Verification and Validation (IV&V)	▪ Conducts IV&V assessments. Identifies potential improvements or identifies problems before they occur ▪ Reviews the testing evidence for the enterprise application by inspecting test cases, test execution results and test reports of every testing activity
Subject Matter Experts (SME) from NM HSD	▪ Provides business knowledge during the preparation of test cases and approval ▪ Responsible for participating in UAT
HSD Requirements Manager	▪ Represent the State as the point person for Requirements Management ▪ Liaison with the HSD Test Manager

Role	Responsibility
HSD Test Manager	▪ Maintain the MMISR Test Management lifecycle ▪ Act as the point person for HSD and the module contractors for test management ▪ Ensure that there is business representation within the testing process ▪ Collaborate with the EPMO, IT Project Managers, BAs and the Module Project Managers in ensuring adherence to this plan and supporting any updates ▪ Collaborate with the MMISR Requirements Manager to support requirement traceability ▪ Collaborate with the MMISR Security Manager to support the security requirement needs of the project ▪ Collaborate with the MMISR Certification Manager to support the certification testing needs of the project ▪ Identify End-to-End testing approach and process
HSD UAT Test Manager	▪ Support the MMISR Test Management lifecycle ▪ Act as the point person for HSD and the module contractors for HSD test management efforts from UAT forward ▪ Collaborate with the EPMO, IT Project Managers, BAs and the Module Project Managers in ensuring adherence to this plan and supporting any updates ▪ Develop test plans ▪ Oversee the HSD test team for UAT case development and UAT test execution
MAD Tester	▪ Support the MMISR Test Management lifecycle ▪ Develop and review test cases ▪ Execute test cases ▪ May participate in UAT testing phase or provide guidance to UAT Testers ▪ Participate in operational readiness testing
HSD Tester	▪ Support the MMISR Test Management lifecycle ▪ Contribute to the development of test plans ▪ Develop and review test cases ▪ Execute test cases ▪ Execute test cases in UAT testing phase ▪ Collaborate with MAD business users to complete testing
HSD Security Tester	▪ Support the MMISR Test Management lifecycle ▪ Develop test plans ▪ Develop test cases ▪ Execute test cases ▪ Liaison with Module Contractor Security and ITD Security for all testing ▪ Perform vulnerability scans on Module Contractor environments ▪ Review Module Contractor test reports ▪ Enforce compliance with the HSD Security standards

Role	Responsibility
HSD Information Technology (IT) PM for Module Contractor Oversight	▪ Ensure the timeliness of HSD activities associated with the testing processes ▪ Escalation point for issues within the testing processes: As necessary, break-out meetings can be scheduled and facilitated by EPMO, if appropriate, but should be jointly planned with HSD PM to address the specific escalated issue. ▪ A report out on progress on the escalated issue will be expected at the next Stand-up meeting will be expected ▪ Prepare for and attend CCB, Technical Change Review Board (TCRB), HSD Data Owners (Part of DGC), and Architecture Review Board (ARB) meetings when changes identified by the testing process are being considered for change ▪ Enforce compliance with the TMP
HSD Module Owners	▪ Participate in daily or weekly issue resolution meetings ▪ Escalation point for blockers within the testing processes ▪ Attend CCB, Weekly Risk and Issues meeting, Bi-Monthly MMISR Module Contractor status meeting, HHS 2020 Status meeting, Technical Change Review Board (TCRB), HSD Data Owners (Part of DGC), and Architecture Review Board (ARB) meetings when changes identified by the testing process are being considered for change ▪ Support all staff with compliance to the TMP
Contract Managers	▪ Participate in daily or weekly issue resolution meetings ▪ Escalation point for blockers within the testing processes related to contractual obligations and SOWs ▪ Attend CCB, Weekly Risk and Issues meeting, Bi-Monthly MMISR Module Contractor status meeting, HHS 2020 Status meeting, Technical Change Review Board (TCRB), HSD Data Owners (Part of DGC), and Architecture Review Board (ARB) meetings when changes identified by the testing process are being considered for change ▪ Support compliance with TMP by providing insight into contract deliverable requirements
MMISR Project Director	▪ Module teams will escalate issues to MMISR Project Director through the MMISR Stand-up Meeting as the first opportunity to escalate issues ▪ A report out on progress on the escalated issue will be expected at the next Stand-up meeting ▪ If necessary, the MMISR Project Director will escalate to Leadership for resolution

5.0 Assumptions / Constraints / Risks

5.1 Assumptions

The following assumptions have been identified for the MMISR test management approach and plan:

- NM HSD resources will be available to participate in the discussions, follow up meetings, and email conversations that are needed to support the testing deliverables in the identified timeline for the deliverable
- NM HSD owns and maintains all the tools used for the implementation of HHS2020 (except for module contractor tools provided through their solutions)

- NM HSD Tools Governance Committee assesses, and plans changes to the tools leveraged to support the project
- Any deviations by any module contractors in using this toolset for commonly identified tasks require proper justification and approval from the ARB and, if data related, the HSD Data Owners (Part of DGC)
- All participating Module Contractors will adopt the MMISR testing methodology/phases as appropriate. Any deviations in adopting the testing phases shall be documented in an addendum to this plan
- The infrastructure of environments needs to be installed and configured before any development and testing activities start
- MMISR Module Contractors will be available to perform integration testing in the appropriate environment and provide support until the application is deployed in production and thereafter
- All participating module contractors will adhere to the applicable State and Federal standards.
- All participating modules in the HHS 2020 enterprise solution will adhere to the security standards specified in the SSP and SDP.
- For purposes of the TMP, the SI is included in the Module Contractors. All responsibilities where module contractor is listed as the responsible party, includes SI as well as BPO Module Contractors
- The use of the term “Quality Assurance” and “QA” refers to the work performed during QAT and/or QA testing and does not refer to the HHS2020 Quality Assurance Module Contractor

5.2 Constraints

The following constraints have been identified for the overall test management approach and plan:

- Subject Expertise: As the HHS2020 enterprise solution components are API driven, MAD testers need to have at least the minimum subject matter expertise required to verify and validate the business workflows
- COTS Products: Unlike custom built components, COTS products come with their own restrictions, limiting interoperability among integrating modules
- System Integration: Use of mocking mechanisms is required, as the modules are being developed in different timelines and on separate schedules. This will delay the effective testing of the enterprise solution until the SIT stage
- Technology: The testing team needs to possess a diverse skill set due to the multiple technologies and tools used for development of enterprise system

5.3 Risks

The Risk Management Plan describes the management of project risks and updates for all areas of the project including the TMP.

The risk register is located on SharePoint.

There are no open or known risks related to the TMP.

The following are an initial set of risks identified for testing the HHS2020 enterprise solution. These risks primarily arise from the constraints listed in Subsection 5.2 above.

- **Scope Risks:** There may be an increase in scope, or a change to the requirements when new Module Contractors come onboard and gather requirements for their modules. Onboarding

and timelines of different Module Contractors poses risk to the integration testing efforts and timeline, as well as regression testing

- **Technology Risks:** Each of the interfacing partners and agencies are constrained by their native technology stacks that will partially or completely restrict interfacing with them
- **Schedule Risks:** The integration and system testing of the HHS2020 enterprise solution will depend on the schedule of all integrating contractor systems. Any delay in the integration will impact the overall project schedule
- **Inconsistency Risks:** Inconsistency in individual Module Contractors testing practices or tools usage can lead to difficulties in effectively managing the testing process
- **Resource Risks:** Since the duration of HHS2020 enterprise solution implementation is long, there is a risk in managing resource churn which could impact the steady state of the project

5.4 Issues

The Issue Management Plan describes the management of project issues and updates for all areas of the project including the TMP.

The issue register is located on SharePoint.

There is no open or known issue related to the TMP.

6.0 CMS Certification

Demonstrating testing compliance with the CMS certification requirements, Medicaid Enterprise Certification Toolkit (MECT) and Outcomes Based Certification (OBC) are an essential part of the overall MMISR Project. The MMIS Certification process ensures that the MMISR enterprise solution and its related modules meet CMS Certification requirements, including but not limited to:

- MECT or OBC
- Artifacts
- Project deliverables

The following items are validated as part of test plan management:

- Testing strategy
- System test plan deliverables and artifacts/evidence
- System test reports and artifacts/evidence
- Acceptance test plan deliverables and artifacts/evidence

MECT and OBC requires the following formal reviews:

- Operational Readiness Review
- Certification Milestone Review

For the Operational Readiness Review, RTM, fully functional backwards and forward traceability, is a required artifact for certification. The test cases and test results for each testing type are mapped to the associated requirement(s). This provides traceability from the identification of the requirement, through design, testing, and implementation. The selective test cases and test results for each testing type are mapped to the associated certification requirements and provided as part of the RTM. An additional requirement is the production of certification evidence for the final certification review.

For the Certification Milestone Review R3, certification is performed using production data and related artifacts and showing evidence from production systems. The evidence for business criteria could include examples of correctly executed use cases and associated operational transactions from the

production system(s) as well as database queries or reports showing correct results for each case. If evidence contains PII or PHI, the evidence will be protected using appropriate measures. In all cases, concrete evidence needs to be provided, as appropriate to the final certification review.

Table 8 is a mapping of this documents to the CMS Testing Framework.

Table 8 - Cross Reference from Plan to CMS Template

Section in CMS Template (CMS Testing Framework)	Section in Plan
1.1 Purpose	1.1 Test Management Plan Purpose
1.2 Scope	1.2 Test Management Plan Scope
1.3 Audience	1.2 Test Management Plan Scope (paragraph 3)
2.1 Business Application Testing Overview	2.1 Business Application Testing Overview
2.2 Infrastructure Testing Overview	2.2 Infrastructure Testing Overview
2.3 Categories of Testing	Appendix E: CMS Categories of Testing
2.5 Reviews	2.6 Test Readiness Reviews 7.0 Certification
2.6 Roles and Responsibilities	2.6.1 Test Readiness Roles and Responsibilities 2.7.1 Test Data Roles and Responsibilities 5.0 Roles and Responsibilities
2.7 Testing Tools	10.4 Appendix D: Testing Tools
2.8 Test Data	2.7 Test Data and Simulation 2.8 Test Data Design Methodology
3 Development Testing	10.5.1 Development Testing
4 Validation Testing	10.5.2 Validation Testing
5 Implementation Testing	10.5.3 Implementation Testing
6 Operational Testing	10.5.4 Operational Testing

7.0 Deliverable Development

7.1 Deliverable Acceptance Criteria

The table below lists the Deliverable Acceptance Criteria:

Table 9 - Deliverable Acceptance Criteria

Item #	Description
1	Deliverable meets quality checklist, including Deliverable Standards check list items
2	Deliverable meets requirements and description of the contract Statement of Work
3	Deliverable meets the details of this DED
4	Deliverable meets CMS guidance

8.0 Appendices

8.1 Appendix A: Deliverable Record of Changes

The deliverable will include a record of changes in the following form:

Table 10 - Deliverable Record of Changes

Version Number	Date	Author/Owner	Description of Change
V0.1	08/15/18	SI Testing Team	DED
V0.2	09/11/18	SI Testing Team	Draft
V0.3	09/28/18	SI Testing Team	Draft with comments addressed
V0.3	09/28/18	SI Testing Team	Final Plan
V0.4	04/01/20	EPMO	Updates based on approved addendum of 7/15/19 and needed changes
V1.0	7/15/2020	EPMO	Updates based upon UAT approach outlined by NM HSD UAT Test Manager in 06/2020
V1.1	8/12/2020	EPMO	Updates based upon HSD and IV&V reviewer feedback
V1.2	9/15/2020	EPMO	Updates based upon HSD and IV&V reviewer feedback
V1.3	10/1/2020	EPMO	Final Submission
V1.4	10/2/2020	EPMO	Final updates to Table 21 and MMISR Testing phase after Deliverable Owner final review
V1.5	TBD	EPMO/K Stevenson/S D'Andrea/Grace A	Clarification for section 508 tools and processes
V1.6	6/23/2022	Karin Stevenson	Annual Update

8.2 Appendix B: List of Acronyms

A list of project-specific acronyms will be maintained on the MMISR SharePoint site:

Table 11 - List of Acronyms

Acronym	Definition
API	Application Program Interface
ARB	Architecture Review Board
ATP	Acceptance Test Plan
BA	Business Analyst
BPO	Business Process Outsourcing
BVT	Breadth Verification Testing
CCB	Change Control Board
CCMP	Change Control Management Plan
CMS	Centers for Medicare and Medicaid Services
COTS	Commercial off-the-shelf
CPU	Central Processing Unit
DoIT	Department of Information Technology
DR	Disaster Recovery
DRP	Disaster Recovery Plan
ESS	Enterprise Shared Services
FedRAMP	Federal Risk and Authorization Management Program
FTI	Financial Transaction Information
HHS	Health and Human Services
HSD	Human Services Department
ICD	Interface Control Document
ICT	Information and Communication Technology

Acronym	Definition
ITD	Information Technology Division
IV&V	Independent Verification and Validation
JAWS	Job Access with Speech
MAD	Medical Assistance Division
MARS-E	Minimum Acceptable Risk Standards for Exchanges
MC	Module Contractor
MECL	Medicaid Enterprise Certification Life Cycle
MECT	Medicaid Enterprise Certification Toolkit
MITA	Medicaid Information Technology Architecture
MMIS	Medicaid Management Information System
MMISR	Medicaid Management Information System Replacement
NM	New Mexico
OBC	Outcomes Based Certification
OM	Operations and Maintenance
PHI	Protected Health Information
PII	Personally Identifiable Information
PMBOK	Project Management Body of Knowledge
PMO	Project Management Office
QA	Quality Assurance
QAT	Quality Assurance Testing
QMP	Quality Management Plan
REST	Representational State Transfer
RMP	Requirements Management Plan
RTM	Requirements Traceability Matrix
SDLC	Software Development Life Cycle
SDP	Security Design Plan
SI	System Integrator
SIT	System Integration Testing
SME	Subject Matter Expert
SMR	System Migration Repository
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SRC	System Review Criteria
SSP	System Security Plan
ST&E	Security Testing & Evaluation
STP	Security Test Plan
STLC	Software Testing Life Cycle
TMP	Test Management Plan
TRR	Test Readiness Review
UAT	User Acceptance Testing

8.3 Appendix C: Referenced Documents

The following is a list of documents references in this plan. Access to the links are based on SharePoint permissions.

Table 12 - Referenced Documents

Document	Link
Requirements Management Plan	Requirements Management Plan
Risk Management Plan	Risk Management Plan
Requirements Traceability Matrix	Requirements Traceability Matrix
Data Validation Process	Data Development and Validation Process (draft)
Defect Management Process	MMISR Project Defect Management Appendix_final
Change Control Management Plan	Change Control Management Plan
System Security Plan	System Security Plan
Enterprise Project Schedule	Enterprise Project Schedule
PMO20 Release Strategy	Release Strategy
CMS XLC Phases	CMS XLC Phases
Governance Council - ARB	ARB Charter
Governance Council - BTC	BTC Charter
Governance Council - CCB	CCB Charter
Governance Council - DGC	DGC Charter
Governance Council - MMISR PMO	MMISR Charter

8.4 Appendix D: Testing Tools

The master list of approved tools and licenses is procured and maintained by NM HSD. Usage details can be found in the table below:

Table 13 - Testing Tools List

Category	Tool	Usage
Test Case Management	Xray for Jira	Xray for Jira is a test management add-on for Jira. Xray supports both manual and automated tests and helps to manage the complete testing life cycle that includes test planning, test designing, and test execution
Requirements Management	Jama	JAMA is the requirements management software offering requirements documentation and approval, traceability, and test management
Defect Management	Jira	Jira is a defect a management tool that allows the creation and tracking the defects
Web Service Simulation/Performance Testing	SoapUI Pro /	SoapUI Pro is a web service testing tool for Service-Oriented Architecture (SOA) and Representational State Transfers (REST). Its functionality covers web service inspection, invoking, development, simulation and mocking, functional testing, and load testing
Performance Testing	LoadUI (ReadyAPI)	LoadUI is a load testing software, targeted mainly at web services. LoadUI allows users to test the speed and scalability of APIs, preview API performance behaviors before releasing to production environments
REST Application Program Interfaces (APIs) Testing	Postman	Postman is a tool used to test REST APIs. Postman allows the creation of collections of integration tests to ensure that APIs are working as expected
Monitoring	Splunk	Splunk is a tool to collect, monitor, and analyze logs generated by applications and databases

8.5 Appendix E: CMS Categories of Testing

As noted, CMS defines four (4) categories of testing: Development Testing, Validation Testing, Implementation Testing, and Operational Testing. These categories are customized to suit the phases of the MMISR project enterprise life cycle and to ensure that it meets stated requirements. Please note, all tables containing inputs, outputs, entry and exit criteria, and process activity steps are meant as examples for each type of process activities needed. Final process activity tables will be defined and updated specific to the Module Contractor's offered solution(s) and will be reviewed and approved by NM HSD. Testing reports to show evidence of test execution will be defined by module contractors, aligned to each Module Contractor's SOW, and will be subject to HSD review and approval.

8.5.1 Development Testing

Development testing is a set of test functions performed within the development environment. The objective of development testing is to verify the status of development. Development testing is the initial testing phase where defects are identified and addressed as early as possible in the process.

The following testing phases are performed by the development team before promoting the build to the SIT environment for further testing:

8.5.1.1 Infrastructure Testing

In the development testing phase, infrastructure testing focuses on validating the hardware and software platforms in the development environment. The objective of infrastructure testing includes validating the hardware installation, network connectivity, network access, software platform installation, and configuration, as well as its' readiness for development use.

The input/output, entry/exit criteria, and the process activities involved in the infrastructure testing can be found in the table below.

Table 13 - Development - Process Activities of Infrastructure Testing

Entry Criteria	Input
▪ The development environment is set up and configured according to the system design and installation plan ▪ The necessary software platform is installed and configured ▪ Infrastructure testing is performed during initial software rollout and every time there is a change in the infrastructure, such as the addition of an integrating system or an upgrade to the existing infrastructure as determined by the CCMP	▪ Baselined Infrastructure and Environments System Test Plan
Process Activity	
▪ Testers create test cases and test sets pertaining to infrastructure testing according to the System Design and Installation Plans ▪ The test plan and test cases are developed based on the hardware and network blueprint, specifications of memory utilization, scalability and security of the infrastructure, and all necessary software components ▪ Test cases are reviewed by infrastructure engineers, QA leads, and developers ▪ Infrastructure test cases are executed by infrastructure engineers ▪ Test results are compared with the expected results. If any discrepancies are found, a defect is created in Jira against the failed test case for resolution and traceability. Retest, as required ▪ Testing reports of infrastructure test execution are prepared based on the test results	

▪ Test reports are reviewed by infrastructure engineers, security leads, QA leads, developers, and project managers and uploaded to SharePoint	
Exit Criteria	**Output**
▪ Successful validation of development environment infrastructure	▪ Development environment infrastructure testing reports

8.5.1.2 Unit Testing

Unit testing is performed by the developer after, or in parallel with, initial application development. It can be performed later as changes are made to the software unit. The developer establishes test cases (in terms of inputs, outputs, expected results, and evaluation criteria) and test data for each software unit.

The following are the objectives of unit testing:

- Unit testing identifies and corrects any internal logic errors in the software units at an early stage of software development
- Unit testing facilitates the tracing of the fault or failure of the software unit by validating the logic and algorithm
- HSD expects the Module Contractor to perform exhaustive unit testing to ensure adequate test coverage

The input/output, entry/exit criteria, and the process activities involved in the unit testing can be found in the table below.

Table 14 - Development - Process Activities of Unit Testing

Entry Criteria	**Input**
▪ Requirements and designs related to the software unit to be tested are identified and analyzed	▪ Analysis of design and algorithm involved in the software unit ▪ Tools identified for unit testing
Process Activity	
▪ Developers identify a set of unit tests covering path testing, boundary condition testing, and input validation and syntax testing of the software units ▪ Test data (mock data) is prepared according to the unit test ▪ Test cases and the test data are reviewed by peer developers and leads ▪ The unit test is performed, and test results are compared with the expected results. If discrepancies are found, reasons for the failure are analyzed and reworked, as required ▪ Code coverage is performed, and the metrics are documented for continuous process improvement ▪ Code coverage is exempted in case the underlying technology stack (COTS) does not support it ▪ Teardown of test data used is performed to make sure test can be repeated with the same data as needed	
Exit Criteria	**Output**
▪ Successful completion of unit tests	▪ Unit test reports, code coverage (as applicable), and quality check metrics

8.5.1.3 Unit Integration Testing

Unit Integration testing focuses on integration testing of the individual software unit with one or more internal and external systems/modules/components/services/interfaces in the development environment. Unit integration testing is performed to validate each component's ability to meet its

stated requirements and to ensure interoperability of the major software components. In the absence of a real partner for integration, mock services and mock data is used to perform the unit integration.

The objective of unit integration testing is to validate the incremental integration of each software unit into a complete software component.

The input/output, entry/exit criteria, and the process activities involved in the unit integration testing can be found in the table below.

Table 15 - Development - Process Activities for Unit Integration Testing

Entry Criteria	**Input**
▪ Requirements and system designs related to the software unit to be integration tested are identified and analyzed ▪ Unit tests are successful and ready for integration test ▪ Unit integration testing is performed after a thorough analysis of whether the software unit being developed needs to integrate with other systems	▪ Baselined system-level design document and Interface Control Documents (ICDs) ▪ Unit integration test case and test data (mock) ▪ Tools identified for unit integration testing
Process Activity	
▪ A set of unit integration tests cases is created and executed by developers based on the system design and ICDs ▪ Mock services are used in place of missing component/module required for integration ▪ Test data (mock data) is prepared according to the test cases developed ▪ Peer reviews of test cases and the test data are performed by developers and the leads ▪ Unit integration tests are executed as per the test cases and test results are compared with expected results. If discrepancies are found, reasons for the failure are analyzed and reworked, as required ▪ Teardown of test data used will be performed to make sure test can be repeated with the same data as needed	
Exit Criteria	**Output**
▪ Successful completion of unit integration tests ▪ QA team reviews the unit integration reports and qualifies the build for further QAT validation	▪ Eligible build from development testing phase to QAT validation ▪ Unit/Integration test results and code analysis and coverage metrics ▪ Unit integration test results are expected from all parties involved in the integration

8.5.2 Validation Testing

Validation testing is a set of test functions performed. Validation testing will help determine that requirements (e.g. functional, security, performance, and infrastructure) are met, and that relevant systems and data can meet the requirements of the MMISR Project. Validation testing is performed by the testing team during the following project phases:

- Infrastructure Testing
- Breadth Verification Testing (BVT) / Smoke Testing
- Functional Testing
- Security Testing
- Performance Testing
- Integration Testing

- Regression Testing
- System Testing

Further, the Data Validation process is being established for the MMISR project. The Data Management Workgroup has reviewed the process in draft form. Validation testing that is completed will follow and include all aspects of the data validation process. A link to the process is included in Appendix C.

8.5.2.1 Infrastructure Testing

Assesses hardware installation, network connectivity, network access, software platform installation, and configuration, as well as its readiness check for application use. The input/output, entry/exit criteria, and the process activities involved in the validation testing can be found in the table below.

Table 16 - Validation - Process Activities for Infrastructure Testing

Entry Criteria	Input
▪ DEV, SIT, UAT, and Production environments are set up and configured according to the system design and installation plan ▪ The necessary software platform is installed and configured ▪ Infrastructure testing is performed during initial software rollout and every time there is a change in the infrastructure, such as the addition of an integrating system or an upgrade to existing infrastructure, etc., as determined by the CCMP	▪ Baselined Infrastructure and Environments System Test Plan
Process Activity	
▪ QA testers shall create test cases and test sets pertaining to infrastructure testing according to the System Design document and the Installation Plan ▪ The test plan and test cases are developed based on the hardware and network blueprint, specifications of memory utilization, scalability, the security of the infrastructure, and all necessary software components ▪ Infrastructure engineers and QA leads review the test cases ▪ Infrastructure engineers execute the infrastructure test cases ▪ Test results are compared with the expected results. If any discrepancies are found, a defect is created in Jira against the failed test case for resolution and traceability ▪ Testing reports for infrastructure test execution are prepared based on the test results ▪ Test reports are reviewed by infrastructure engineers, QA leads, and project managers	
Exit Criteria	**Output**
▪ Validation of SIT environment infrastructure according to the infrastructure test plan and ready for functional testing with no outstanding critical/major issues	▪ SIT environments infrastructure testing reports

8.5.2.2 Breadth Verification Testing (BVT) / Smoke Testing

Assesses that the core functionalities of the business solution are performed satisfactorily. This is the first step in every testing phase after the build is deployed on the target environment. The input/output, entry/exit criteria, and the process activities involved in the validation testing can be found in the table below.

Table 17 - Validation – Process Activities for Breadth Verification Testing

Entry Criteria	Input
▪ Build deployed on SIT environment	▪ Baselined system test plan

▪ Development testing is completed successfully	
Process Activity	
▪ QA testers create test cases and test sets for the core functionalities of the business solution ▪ QA leads review test cases ▪ Test cases are executed, and the results are compared with the expected results. If any discrepancies are found, a defect is created in Jira against the failed test case for resolution and traceability. Retest, as required ▪ Testing reports are prepared based on the test results ▪ Test reports are reviewed by QA leads and project managers	
Exit Criteria	**Output**
▪ Verification of BVT/Smoke test execution and reports are as per test plan ▪ No critical issues are open, and a viable workaround is identified for major defects	▪ BVT/Smoke test reports and summary ▪ *This TMP will be updated to reflect new SI Module Contractor once onboarded, and test reports can be offered

8.5.2.3 Functional Testing

Assesses the input/output functions of the business application against pre-defined functional and data requirements. The input/output, entry/exit criteria, and the process activities involved in the validation testing can be found in the table below.

Table 18 - Validation - Process Activities for Functional Testing

Entry Criteria	**Input**
▪ The SIT environment is set up and validated using the approved Infrastructure and Environments System Test Plan	▪ Baselined functional and business requirements document ▪ System design document ▪ Baselined system test plan ▪ Tools identified for functional testing
Process Activity	
▪ Testers identify functions of the application and document the test cases and tests sets ▪ Mock services are used in place of missing components/modules required for integration ▪ Testers prepare (mock) the test data for functions that require simulation behavior ▪ Test Leads review test cases ▪ BVT/Smoke, regression, performance, and security testing are performed as part of functional testing, as required ▪ Test cases are executed, and the results are compared with the expected results. If any discrepancies are found, a defect is created in Jira against the failed test case for resolution and traceability. Retest, as required ▪ Testing reports are prepared based on the test results ▪ Test leads and project managers review test results	
Exit Criteria	**Output**
▪ Verification that functional test execution and reports are as per test plan ▪ No critical issues are open and viable workarounds are identified for major defects	▪ Test case execution results ▪ Test summary report

8.5.2.4 Security Testing

Assesses that the application is free from any security vulnerabilities and verifies that unauthorized user access and confidential data access is prevented. The input/output, entry/exit criteria, and the process activities involved in the validation testing can be found in the table below.

Table 19 - Validation - Process Activities for Security Testing

Entry Criteria	Input
▪ Baselined System Test Plan for Security ▪ Security Testing will be performed for every major release. It is also performed for releases where there are potential software changes that may cause security risks, as determined by the CCMP ▪ The testers are trained on the security standards and tools	▪ Baselined System Security Test Plan ▪ Tools identified for security testing
Process Activity	
▪ Testers, specialized in security testing, create the System Security Test Plan covering the approach, roles and responsibilities, tools, and schedule for the security testing ▪ Testers create and execute security test cases based on the approved System Security Test Plan ▪ Testers compare the test results with the expected output. If discrepancies are found, a defect is created in Jira against the failed test case for resolution and traceability. Retest as required ▪ The Test lead publishes the Security Test Results based on the test cases executed and results are summarized ▪ The Test leads, developers, and project managers review the reports	
Exit Criteria	**Output**
▪ Validation that the HHS 2020 MMISR enterprise application addresses the Minimum Acceptable Risk Standards for Exchanges (MARS-E) 2.0 controls	▪ System Security Test Report

8.5.2.5 Performance Testing

Assesses the capacity and throughput of a business application and/or infrastructure in processing time, CPU utilization, network utilization, and memory and storage capacities relative to expected normal (average and peak) user and processing load. The input/output, entry/exit criteria, and the process activities involved in the validation testing can be found in the table below.

Table 20 - Validation - Process Activities for Performance Testing

Entry Criteria	Input
▪ Successfully completed the BVT/Smoke testing in functional, integration, and system testing phases ▪ Performance testing is conducted, as required, for major releases and for changes that can potentially impact the performance of the application, as determined by the CCMP ▪ The testers are trained on the performance testing tools	▪ Baselined performance test plan ▪ Tools identified for performance testing
Process Activity	
▪ Testers, specialized in performance testing, create the performance test plan covering the approach, roles and responsibilities, tools, and schedule for the performance testing ▪ Testers create and execute performance test cases and prepare (mock) the test data according to the test cases developed ▪ Testers compare the test results with the expected output. If discrepancies are found, a defect is created in Jira against the failed test case for resolution and traceability. Retest, as required.	

▪ Test lead publishes the Performance Test Results based on the test cases executed and the results are summarized ▪ Reports are reviewed with lead testers and developers, solution architects, and project managers ▪ Testers will update the test cases once the performance benchmark is set and will use the benchmark reading for future performance testing	
Exit Criteria	**Output**
▪ Verified that performance test execution and reports are as per test plan ▪ No critical issues are open and viable workaround is identified for major defects	▪ Test case execution results ▪ Test summary report

8.5.2.6 Integration Testing

Focuses on the integration of the individual system with one (1) or more internal and external systems/modules/components/services/interfaces. The input/output, entry/exit criteria, and the process activities involved in the validation testing can be found in the table below.

Table 21 - Validation – Process Activities for Integration Testing

Entry Criteria	**Input**
▪ Successful completion of functional testing	▪ Baselined Integration Test plan ▪ System design documents and ICDs ▪ Tools identified for integration testing
Process Activity	
▪ Integration test cases and test sets are created using system design and ICDs ▪ Test cases are reviewed by the Test leads for all integrating systems ▪ BVT/Smoke, regression, performance, and security testing are performed as part of the integration testing, as required by all integrating systems ▪ Test cases are executed, and the results are compared with the expected results. If any discrepancies are found, a defect is created in Jira against the failed test case for resolution and traceability. Retest, as required ▪ Testing reports are prepared by Test leads based on the test results ▪ Test reports are reviewed by Test leads and project managers of all integrating systems	
Exit Criteria	**Output**
▪ Verification that the integration test execution and reports are as per test plan ▪ No critical issues are open and viable workarounds are identified for major defects	▪ Test case execution results ▪ Test summary report

8.5.2.7 Regression Testing

Performed on a build to confirm that a recent change has not adversely affected preexisting and validated functionalities or introduced new behaviors that are undesirable. Regression testing includes retesting the previously failed tests to ensure that no other issues are caused. The input/output, entry/exit criteria, and the process activities involved in the validation testing can be found in the table below.

Table 22 - Validation - Process Activities for Regression Testing

Entry Criteria	**Input**
▪ Successfully completed the BVT/Smoke test in functional, integration, and system testing phase	▪ Approved regression test cases.

Process Activity	
▪ Regression test cases are executed, and the results are compared with the expected results. If any discrepancies are found, a defect is created in Jira against the failed test case for resolution and traceability. Retest, as required ▪ Regression testing includes retesting the previously failed test to ensure that the issues are now fixed ▪ Testing reports are prepared based on the test results ▪ Test leads and project managers review test reports ▪ Testers update the new functionality test cases in the regression test suite	
Exit Criteria	**Output**
▪ Verified that regression test execution and reports are as per test plan ▪ No critical issues are open, and a viable workaround is identified for major defects	▪ Regression test reports and summary

8.5.2.8 System Testing

An end-to-end functional validation with interfaces using pre-defined integration test case and test data. The input/output, entry/exit criteria, and the process activities involved in the validation testing can be found in the table below.

Table 23 - Validation - Process Activities for System Testing

Entry Criteria	Input
▪ Successful completion of functional and integration testing in the QAT environment ▪ Successful approval of the TRR1 report	▪ Baselined System Test Plan ▪ Baselined functional and business requirements document ▪ System Design document
Process Activity	
▪ System test cases and test sets are created using system design and ICDs covering end-to-end testing for all components/modules/external interface to verify all functional and non-functional requirements ▪ Test cases are reviewed by Test leads of all integrating systems ▪ BVT/Smoke, regression, performance, and security testing are performed as part of system testing, as required, by all integrating systems ▪ Test cases are executed, and the results are compared with the expected results. If any discrepancies are found, a defect is created in Jira against the failed test case for resolution and traceability. Retest, as required ▪ Testing reports are prepared by QA leads based on the test results ▪ Test reports are reviewed by QA leads and project managers of all integrating systems	
Exit Criteria	**Output**
▪ Verification that the system test execution and reports are as per test plan ▪ No critical issues are open and viable workarounds are identified for major defects	▪ Test case execution results ▪ Test summary report

8.5.2.9 Section 508 Testing

Performed to validate that applicable Section 508 Accessibility Standards on the user interface or any electronic output are met.

The input/output, entry/exit criteria, and the process activities involved in the validation testing can be found in the table below. Section 508 testing applies to the Unified Portal (UP), and any of the module contractors BPOs offered solutions. The SI platform itself -- unless an HSD resource needs

508 compliance support -- is not subject to Section 508 testing. This entire section will be updated when the UP Module Contractor and other Module Contractor BPOs are selected.

Table 24 - Validation - Process Activities for Section 508 Testing

Entry Criteria	Input
▪ User interfaces and electronic outputs are developed as per the design and approved ▪ Unit level Section 508 compliance testing performed by the development level is successful ▪ The testing team is trained on Section 508 standards and tools to test accessibility ▪ Section 508 will be performed, as required, for every major release which has a user interface and or electronic output, as determined by the CCMP	▪ User Interface and electronic outputs like reports ▪ Section 508 standards and checklists ▪ Tools identified for Section 508 testing
Process Activity	
▪ QA testers specialized in Section 508 testing create the test cases and checklist to verify the design aspects of elements in the user interface and electronic output ▪ Screen readers are used to ensure interactive interface elements like buttons, menus, web forms, images, and multimedia are readable and accessible by keyboard ▪ Test cases are executed, and the results are compared with the expected results. If any discrepancies are found, a defect is created in Jira against the failed test case for resolution and traceability. Retest, as required ▪ Testing reports are prepared based on the test results ▪ QA leads and project managers view the test reports	
Exit Criteria	**Output**
▪ Verified that Section 508 test execution and reports are as per test plan ▪ No critical issues are open and viable workarounds are identified for major defects	▪ Section 508 test results

8.5.3 Implementation Testing

Implementation testing is a set of testing functions performed within a production-like environment to confirm that the HHS2020 enterprise application operates in accordance with the architectural and technical requirements. This phase is executed on the UAT or production-like environments.

The objective of this phase is as follows:

- To verify and validate that the infrastructure solution behaves as required in UAT and production-like environments, and that it is configured with the same infrastructure as found in the target production environment
- To confirm that the security settings of the business application comply with the System Security Plan

Implementation testing includes the following testing phases:

- Infrastructure Testing
- Security Testing
- Performance Testing
- Contingency Testing
- User Acceptance Testing

8.5.3.1 Infrastructure Testing

Assesses the hardware installation, network connectivity and access, and software platform installation and configuration, as well as acting as a readiness check for production usage. The input/output, entry/exit criteria, and the process activities involved in the implementation testing can be found in the table below.

Table 25 - Implementation - Process Activities for Infrastructure Testing

Entry Criteria	Input
▪ Pre-Prod environments are set up and configured according to the system design and installation plan ▪ The necessary software platform is installed and configured ▪ Infrastructure testing shall be performed during initial software rollout and every time there is a change in the infrastructure, like the addition of an integrating system or an upgrade to existing infrastructure, etc., as determined by the CCMP	▪ Baselined Infrastructure and Environments System Test Plan
Process Activity	
▪ QA testers create test cases and test sets pertaining to infrastructure testing according to the System Design, and the Installation Plan ▪ The test plan and test cases are developed based on the blueprint of hardware and network, specifications of memory utilization, scalability and security of the infrastructure, and all necessary software components ▪ Test cases are reviewed by infrastructure engineers and QA leads ▪ Infrastructure test cases are executed by infrastructure engineers ▪ Test results are compared with the expected results. If any discrepancies are found, a defect is created in Jira against the failed test case for resolution and traceability ▪ Testing reports of infrastructure test execution shall be prepared based on the test results ▪ Test reports are reviewed by infrastructure engineers, QA leads, and project managers	
Exit Criteria	**Output**
▪ Verified that Infrastructure test execution and reports are as per test plan ▪ No critical issues are open, and a viable workaround is identified for major defects	▪ Production-like and UAT environments infrastructure testing reports

8.5.3.2 Security Testing

Verifies and validates that the processes, business application, software platform, and infrastructure comply with the MARS-E security controls. The input/output, entry/exit criteria, and the process activities involved in the implementation testing can be found in the table below.

Table 26 - Implementation - Process Activities for Security Testing

Entry Criteria	Input
▪ Baselined System Test Plan for Security ▪ Security testing will be performed for every major release. It is also performed on the releases where there are potential software changes that may cause security risks, as determined by the CCMP ▪ The QA testers are trained on the security standards and tools	▪ Baselined System Security Test Plan ▪ Tools identified for security testing
Process Activity	
▪ QA testers specialized in security testing create the System Security Test Plan covering the approach, roles and responsibilities, tools, and schedule for the security testing	

▪ QA testers create and execute security test cases based on the approved System Security Test Plan ▪ QA testers compare the test results with the expected output. If discrepancies are found, a defect shall be created in Jira against the failed test case for resolution and traceability. Retest, as required ▪ QA lead publishes the Security Test Results based on the test cases executed and results are summarized ▪ Review the reports with QA leads and developers, solution architects, and project managers	
Exit Criteria	**Output**
▪ Validation that the HHS2020 MMISR enterprise application addresses the MARS-E 2.0 controls	▪ System Security Test Report

8.5.3.3 Performance Testing

Involves load and stress testing of the business application to assess that the application and the underlying platform is capable of handling the surge in requests in the production environment. The input/output, entry/exit criteria, and the process activities involved in the implementation testing can be found in the table below.

Table 27 - Implementation - Process Activities for Performance Testing

Entry Criteria	**Input**
▪ Performance testing shall be conducted, as required, for major releases and for changes that can potentially impact the performance of the application, as determined by the CCMP ▪ The QA testers are trained on the performance testing tools	▪ Baselined performance test plan ▪ Tools identified for performance testing
Process Activity	
▪ QA testers specialized in performance testing create the performance test plan covering the approach, roles and responsibilities, tools, and schedule for the performance testing ▪ QA testers create and execute performance test cases and prepare (mock) test data according to the test case developed ▪ QA testers compare the test results with the expected output. If discrepancies are found, a defect shall be created in Jira against the failed test case for resolution and traceability. Retest, as required ▪ QA lead publishes the performance test results based on the test cases executed and results are summarized ▪ Review the reports with lead testers and developers, solution architects, and project managers ▪ QA testers will update the test cases once the performance benchmark is set and will use the benchmark reading for future performance testing	
Exit Criteria	**Output**
▪ Verified that performance test execution and reports are as per test plan ▪ No critical issues are open, and a viable workaround is identified for major defects	▪ Performance Test case execution results ▪ Performance Test summary report

8.5.3.4 Contingency Testing

A critical test function performed before the application is ready for production use and will be performed in a pre-production or production-like environment. The Initial Contingency Planning Testing ensures that the notification and activation requirements and procedures are followed in a timely manner during a contingency. The input/output, entry/exit criteria, and the process activities involved in the implementation testing can be found in the table below.

Table 28 - Implementation - Process Activities for Contingency Testing

Entry Criteria	**Input**

▪ Contingency Plan testing for the infrastructure requires an annual functional disaster recovery testing ▪ Contingency plan testing shall be performed during initial software rollout and every time there is a change in the infrastructure, like the addition of an integrating system or an upgrade to existing infrastructure, etc., as determined by the CCMP	▪ DRPs from all participating module contractors
Process Activity	
▪ QA lead validates DRPs from all participating module contractors in the HHS 2020 enterprise solution along with other QA testers ▪ QA lead validates that the DRP satisfies all CMS, Department of Information Technology (DoIT), and NM HSD requirements along with other QA testers ▪ Application support engineers of all participating module contractors will ensure that the DRPs are reviewed and updated annually ▪ Project manager establishes the OM Manual and QA lead, QA testers, application support engineer review and provide input, if necessary	
Exit Criteria	**Output**
▪ Operations and Maintenance Manual adequately covers the operating procedures	▪ Operations and Maintenance Manual

8.5.3.5 User Acceptance Testing

The phase in which the identified acceptance test cases are executed and validated by NM HSD. Module specific test beds and artifacts will be drafted for each Module Contractor. The results of the UAT are documented by NM HSD in the form of UAT Summary Reports. The input/output, entry/exit criteria, and the process activities, as known today, involved in the implementation testing can be found in the table below. Criteria will evolve over the lifecycle and development of the MMISR project, as more module contractors are onboard.

Table 29 - Implementation - Process Activities for UAT Testing

Entry Criteria	Input
▪ Fully defined Acceptance Test Plan (ATP) developed by NM HSD UAT testing team and approved by the ARB.; non-technical release and/or business functionality releases will be approved by BTC ▪ Successful approval of the TRR2 report	▪ TRR2 Report ▪ NM HSD ATP
Process Activity	
▪ Before entering the UAT phase, each participating module contractor will prepare a TRR for NM HSD approval ▪ An approved TRR2 marks the beginning of the UAT phase which will be conducted by HSD for each participating Module Contractor ▪ Each participating module contractor will produce an ATP, including for the infrastructure and environment components, for their offered solutions. The ATP will describe the approach, planning, roles and responsibilities, and the list of final acceptance test cases for the system under test ▪ HSD will develop their own acceptance test plan including UAT test cases or test sets for each module contractor and the integration of their offered solution ▪ SMEs from NM HSD will execute the test cases documented in the ATP and report defects on Jira, if found ▪ Test execution summary shall be prepared to describe the test results for the cases listed in the ATP	
Exit Criteria	**Output**

▪ The business application is validated by NM HSD against the scope baseline ▪ NM HSD documents the test results for the Acceptance Test Summary report	▪ NM HSD Acceptance Test reports ▪ NM HSD Acceptance Test Summary report

8.5.3.6 End to End Testing

The end-to-end testing phase is planned upon completion of the individual module specific tests and is conducted to ensure that the system behaves cohesively and as expected. The end-to-end strategy for the MMISR solution is under development at the time this TMP (PMO14) is being finalized. Once complete, the end-to-end testing strategy will be added to the TMP as an Appendix. A full plan specific to end-to-end testing will be developed and will go through various levels of review and a Leadership review and approval prior to being finalized and executed against for MMISR go-live.

8.5.4 Operational Testing

Operational testing confirms that the HHS2020 enterprise solution is operational in accordance with architectural and technical requirements in the production environment. The objective of the operational phase is to verify the operational integrity, effectiveness, and resilience of the HHS2020 enterprise solution in the production environment.

Operational testing is comprised of the following phases:

- Infrastructure Testing
- Production Readiness Testing
- Operational Security Testing
- Operational Contingency Testing
- Monitoring and Reliability Testing

8.5.4.1 Infrastructure Testing

Assesses the hardware installation, network connectivity, and access, software platform installation and configuration as well as its readiness check for production usage. The input/output, entry/exit criteria, and the process activities involved in the operational testing can be found in the table below.

Table 30 - Operational - Process Activities for Infrastructure Testing

Entry Criteria	Input
▪ Production and DR environments are set up and configured according to the system design and installation plan ▪ The necessary software platform is installed and configured ▪ Infrastructure testing will be performed during initial software rollout and every time there is a change in the infrastructure, like the addition of an integrating system or an upgrade to existing infrastructure, as determined by the CCMP	▪ Baselined Infrastructure and Environments Security Test Plan (STP)
Process Activity	
▪ QA testers, specialized in infrastructure testing, shall create test cases and test sets according to the System Design and the Installation Plan ▪ The test plan and test cases are developed based on the hardware and network blueprints, specifications of memory utilization, scalability and security of the infrastructure, and all necessary software components	

▪ Infrastructure engineers and QA leads review the test cases ▪ Infrastructure engineers execute the infrastructure test cases ▪ Test results are compared with the expected results. If any discrepancies are found, a defect is created in Jira against the failed test case for resolution and traceability ▪ Testing reports of infrastructure test execution shall be prepared based on the test results ▪ Test reports are reviewed by infrastructure engineers, QA Leads, and project managers of all of the integrating systems	
Exit Criteria	**Output**
▪ Verification that Infrastructure test execution and reports are as per test plan ▪ No critical issues are open and viable workarounds are identified for major defects	▪ Production environment infrastructure testing reports

8.5.4.2 Production Readiness Testing

Leverage a BVT/Smoke test suite prepared exclusively for the production readiness check, to confirm that the application has been installed and configured correctly in the production environment and is ready for operational use. The input/output, entry/exit criteria, and the process activities involved in the operational testing can be found in the table below.

Table 31 - Operational - Process Activities for Production Readiness Testing

Entry Criteria	**Input**
▪ Successful installation and configuration of the production environment	▪ Approved production readiness test cases
Process Activity	
▪ Test cases are executed by application support engineers and results are compared with the expected results. If any discrepancies are found, a defect is created in Jira against the failed test case for resolution and traceability. Retest, as required ▪ Testing reports of production readiness test execution shall be prepared based on the test results ▪ Test reports are reviewed by QA leads, developers, project managers of all integrating systems, and NM HSD	
Exit Criteria	**Output**
▪ Verification that production readiness test execution and reports are according to the test plan ▪ No critical issues are open and viable workarounds are identified for major defects	▪ Production readiness test reports and summary

8.5.4.3 Operational Security Testing

Verifies and validates that the processes, business application, software platform, and infrastructure comply with the MARS-E security controls. The input/output, entry/exit criteria, and the process activities involved in the operational testing can be found in the table below.

Table 32 - Operational - Process Activities for Operational Security Testing

Entry Criteria	**Input**
▪ Operational security testing is performed during the initial rollout of the application and whenever there is a significant change to the system, change in security policies, or major security violations	▪ Baselined System Security Test Plan ▪ Tools identified for security testing

▪ Operational security testing shall be performed periodically, as determined by the SSP	
Process Activity	
▪ QA testers, specialized in security testing, create test cases according to the System Security Test Plan ▪ Application support engineers of all participating module contractors and independent security agencies execute the security test cases based on the approved System Security Test Plan ▪ QA testers and developers analyze the test results. If discrepancies are found, a defect is created in Jira against the failed test case for resolution and traceability. Retest, as required	
Exit Criteria	**Output**
▪ Verification that operational security test execution and reports are according to the test plan ▪ No critical issues are open and viable workarounds are identified for major defects	▪ Operational System Security Test report

8.5.4.4 Operational Contingency Testing

A critical test function performed on the application deployed in the production environment and will be performed in a pre-production or production-like environment. The Operational Contingency Planning Testing is performed to assess that the notification and activation requirements and procedures are followed in a timely manner during a contingency. The input/output, entry/exit criteria, and the process activities involved in the operational testing can be found in the table below.

Table 33 - Operational - Process Activities for Operational Contingency Testing

Entry Criteria	**Input**
▪ Operational contingency testing is performed during the initial rollout of the application and annually, as defined in the DRP for participating vendors	▪ DRPs from all participating module contractors
Process Activity	
▪ Infrastructure engineers and application support engineers of all participating module contractors execute the operational contingency test cases and validate high availability and redundancy of the production system. They also validate the switchover of the production environment to the DR environment in case of contingency ▪ QA testers and developers analyze the test results. If discrepancies are found, a defect is created in Jira against the failed test case for resolution and traceability. Retest, as required ▪ QA lead ensures CPs and DRPs are reviewed and updated annually ▪ Project manager establishes the Operations and Maintenance Manual and the QA lead, QA testers, infrastructure engineers, and application support engineer review and provide input, if necessary	
Exit Criteria	**Output**
▪ Verification that operational contingency test execution and reports are according to the test plan ▪ No critical issues are open and viable workarounds are identified for major defects	▪ Operational Contingency Test Report ▪ Operations and Maintenance Manual is updated to address additional procedures discovered in contingency testing if any

8.5.4.5 Monitoring and Reliability Testing

Continuously monitors performance, incidents, and capacity utilization to maintain operational availability of the HHS2020 enterprise solution and infrastructure. The input/output, entry/exit criteria, and the process activities involved in the operational testing can be found in the table below.

Table 34 - Operational - Process Activities for Monitoring and Reliability Testing

Entry Criteria	Input
▪ Operational monitoring and reliability testing are performed after ensuring that the system is deployed and operational in the production environment ▪ Operational monitoring is a continuous process	▪ Baselined Operations and Maintenance (OM) Manual ▪ Tools identified for monitoring and reliability testing
Process Activity	
▪ Application support engineers of all participating module contractors shall validate the operational effectiveness, suitability, and survivability of HHS 2020 enterprise solution ▪ Operational monitoring is a continuous process to monitor performance, incidents, and capacity utilization of the application. This process involves the triggering of timely alerts and notifications in the event a system is operating in an anomalous fashion ▪ Application or infrastructure deployed to a production environment shall be assessed to determine, whether or not, a rollback will be required to the previous production release ▪ QA testers and developers will analyze the findings, and a defect will be created in Jira against the failed test case for resolution and traceability. Retest, as required	
Exit Criteria	**Output**
▪ Verification that production monitoring test execution and reports are according to the test plan ▪ No critical issues are open and viable workarounds are identified for major defects ▪ Qualification of the release to go live for production use	▪ Operational monitoring and reliability test report ▪ Operations and Maintenance Manual is updated to address additional procedures discovered in monitoring and reliability testing, if any