

AHCA Florida Health Care Connections (FX)

<<Insert Project Name Here>>

Testing Management Plan

Version: 300

Date: Month dd, yyyy

Author: [Author]

Submitted To: AHCA FX Project Management





Revision History

DATE	VERSION	DESCRIPTION	AUTHOR
MM/DD/YYYY	001	<<Insert Project Name Here>> Testing Management Plan first draft version	

Modifications to the approved baseline version (100) of this artifact must be made in accordance with the Artifact Management Standards.

Quality Review History

DATE	REVIEWER	COMMENTS
MM/DD/YYYY		



Table of Contents

Section 1	Introduction.....	1
1.1	Purpose	1
1.2	Scope Statement	1
1.3	Goals and Objectives	2
1.4	Referenced Documents	2
Section 2	Roles and Responsibilities	3
Section 3	Assumptions, Constraints, and Risks.....	4
3.1	Assumptions	4
3.2	Constraints	4
3.3	Risks.....	4
Section 4	Testing Approach Strategy	5
4.1	Testing Approach Overview	5
4.2	Testing Framework.....	5
4.3	Testing Expectations and Success Criteria	5
4.4	Readiness Reviews	5
Section 5	Planned Tests	7
5.1	Entrance and Exit Criteria.....	9
5.2	Development Testing.....	15
5.3	Integration Testing.....	15
5.4	System Testing.....	15
5.5	Regression Testing.....	16
5.6	User Acceptance Testing	16
5.7	Performance Testing	16
5.8	Build Verification Testing / Smoke Testing	16
5.9	Operational Readiness Testing	16
Section 6	Test Progression	18
Section 7	Feature Testing	19
7.1	Features To Be Tested.....	19
7.2	Features Not to Be Tested	19



Section 8	Test Cases	20
Section 9	Defect Tracking and Reporting.....	21
9.1	Defect Management	21
9.2	Severity and Priority	22
9.2.1	Severity Categories	22
9.2.2	Priority Categories.....	23
Section 10	Test Environment	24
10.1	Testing Environments.....	25
10.2	Hardware	27
10.3	Software	27
10.4	Test Data	28
10.5	Other Materials	28
10.6	Installation, Testing, and Control	28
10.7	Security.....	28
Section 11	Test Deliverables.....	29
11.1	Test Case-to-Requirements Traceability Matrix	29
11.2	Test Incident Reports	29
11.3	Test Summary Report	29
Section 12	Test Schedule and Milestones	30
Section 13	Test Roles and Responsibilities	31
13.1	Orientation Plan.....	31
Appendices.....		32



Table of Exhibits

Exhibit 2-1: Roles and Responsibilities	3
Exhibit 5-1: Types of Testing	9
Exhibit 5-2: Developmental Testing Entry and Exit Criteria	10
Exhibit 5-4: System Testing Entry and Exit Criteria	11
Exhibit 5-5: Regression Testing Entry and Exit Criteria	12
Exhibit 5-6: User Acceptance Testing Entry and Exit Criteria	13
Exhibit 5-8: Build Verification Testing Entry and Exit Criteria	14
Exhibit 7-1: Features to Be Tested	19
Exhibit 7-2: Features Not to Be Tested	19
Exhibit 9-1: Defect Life Cycle	21
Exhibit 10-1: FX Environments Diagram	25
Exhibit 10-2: FX Environments Description	26
Exhibit 10-3: FX Environment - Testing Crosswalk	27
Exhibit 10-4: Testing Hardware	27
Exhibit 10-5: Testing Software	27
Exhibit 12-1: Testing Milestones	30
Exhibit 13-1: Testing Resources	31



SECTION 1 INTRODUCTION

The Florida Agency for Health Care Administration (AHCA or Agency) is adapting to the changing landscape of health care administration and increased use of the Centers for Medicare and Medicaid Services (CMS) Medicaid Information Technology Architecture (MITA) to improve the administration and operation of the Florida Medicaid Enterprise. The current Florida Medicaid Enterprise includes services, business processes, data management and processes, technical processes within the Agency, and interconnections and touchpoints with systems that reside outside the Agency necessary for administration of the Florida Medicaid program. The current Florida Medicaid Enterprise System (MES) includes the Florida Medicaid Management Information System (FMMIS), Decision Support System (DSS), and other systems operated by different vendors. These systems in the MES interface primarily through the exchange of data files, via Secured File Transfer Protocol. These point-to-point interfaces become more complex and costlier as the number of systems and applications increase. The future of the Florida Medicaid Enterprise integration is to allow Florida Medicaid to secure services that can interoperate and communicate without relying on a common platform or technology.

During the strategic visioning session held on December 13, 2017, the executive team determined that this project should be focused much more broadly than just a FMMIS replacement, indicating that the project should “Transform the Medicaid Enterprise to provide the greatest quality, the best experience, and the highest value in healthcare.”

To articulate this far-reaching scope, the MES Procurement Project was re-named Florida Health Care Connections (FX) in the summer of 2018.

The Agency contracted with the Strategic Enterprise Advisory Services (SEAS) Vendor, in September 2017 to develop the technology standards and propose solutions for FX in accordance with the CMS Conditions and Standards, including MITA 3.0, and to provide strategic, programmatic, and technical advisory services for the Agency. The 17 initial deliverables were accepted by the Agency in Fiscal Year (FY) 2017-18. The SEAS Vendor is now executing to those deliverables and performing the annual refresh as required by the SEAS Contract, MED191.

1.1 PURPOSE

The Testing Management Plan describes the overall technical and management approach, resources, and schedule for all intended test activities associated with development, validation, implementation, and operational testing.

1.2 SCOPE STATEMENT

The Testing Management Plan will provide a comprehensive description of the testing approach and strategy, including:



- Roles and Responsibilities
- Assumptions/Constraints/Risks
- Testing Approach/Strategy
- Planned Tests
- Test Progression
- Feature Testing
- Test Cases
- Defect Tracking and Reporting
- Test Environment
- Test Deliverables
- Test Schedule and Milestones
- Test Roles and Responsibilities

1.3 GOALS AND OBJECTIVES

<Instructions: Identify the goals and objectives for this plan.>

- Goal #1 – The goal of this plan is to <insert language>
 - › Objective #1 – <insert objective>
 - › Objective #2 – <insert objective>
- Goal #2 – The goal of this plan is to <insert language>
 - › Objective #1 – <insert objective>
 - › Objective #2 – <insert objective>

1.4 REFERENCED DOCUMENTS

The following documents were used as input to the development of the Testing Management Plan and provided valuable information to produce the procedures and processes.

- Medicaid Enterprise Certification Toolkit
- Centers for Medicare and Medicaid Services, eXpedited Life Cycle (CMS XLC) Test Plan
- Testing Requirements of finalized FX Project Procurements



SECTION 2 ROLES AND RESPONSIBILITIES

Exhibit 2-1: Roles and Responsibilities identify the roles and responsibilities for the primary stakeholders that maintain or use this document.

<Instructions: Specify each major role (do not name the individual) and the major activities related to this document.>

ROLE	RESPONSIBILITY
	▪
	▪
	▪
	▪
	▪
	▪
	▪
	▪

Exhibit 2-1: Roles and Responsibilities



SECTION 3 ASSUMPTIONS, CONSTRAINTS, AND RISKS

3.1 ASSUMPTIONS

<Instructions: If the testing approach/strategy is based upon any assumptions, list and describe them. For example, identify dependencies with other systems and the assumption that they will be ready to test when needed, assumptions regarding availability of defined test environments, etc.>

3.2 CONSTRAINTS

<Instructions: Describe any limitations or constraints that have a significant impact on the testing of the system, application, or situation. Such constraints may be imposed by any of the following (the list is not exhaustive):

- Hardware or software environment
- End-user environment
- Availability of resources
- Interoperability requirements
- Interface/protocol requirements
- Data repository and distribution requirements.>

3.3 RISKS

<Instructions: Identify and describe the potential problems or risk areas of the project and/or issues which may have an impact upon the testing effort. Some examples might include system interfaces, highly complex software, system load issues, security, performance, and reliability. If any issues arise during the prescribed testing activities that lead to new risks, they should be documented in the project's Risk Report.>



SECTION 4 TESTING APPROACH STRATEGY

4.1 TESTING APPROACH OVERVIEW

<Instructions: Describe the overall approach that will be used to test all functions, features, and requirements of the automated system, application, or situation for which the Test Plan applies. As applicable to this Test Plan, describe the measures to be taken to ensure all aspects of the system are successfully tested and can be implemented. Document key aspects of the testing approach such as content, methodology, prioritization, and progression of development, system integration, regression, performance, user acceptance, and operational readiness testing activities to be performed during the corresponding life cycle phases. Reference the Project Management Plan (PMP) / Development Approach Plan and Project Process Agreement (PPA), as appropriate.>

4.2 TESTING FRAMEWORK

<Instructions: Describe how the testing framework will be applied to the project. For example, will some unit and application integration testing be done, and then some more development, and so on? Is a prototype being built that will be usability tested before the releasable software is developed? Also include plans for testing related documentation (e.g., installation instructions, User Manual, Operations & Maintenance (O&M) Manual, Training Artifacts, etc.) and for conducting applicable readiness reviews. Also, if applicable, describe how reuse will be applied to the testing effort to make testing more efficient and less costly.>

4.3 TESTING EXPECTATIONS AND SUCCESS CRITERIA

<Instructions: Describe testing expectations and success criteria.

4.4 READINESS REVIEWS

<Instructions: Describe readiness reviews that occur related to different types of testing.

The following is boilerplate text regarding readiness reviews conducted for the majority of projects. The specific project modifies the text as appropriate for the given project.>

A Validation Readiness Review (VRR) occurs after completion of all development testing to affirm final agreement from all stakeholders that the <automated system/application> is ready to begin validation testing. The VRR seeks to ensure that all prerequisites leading up to validation testing were met.

<Instructions: Describe project specific validation readiness review prerequisites, review criteria, review processes, communication, or approval processes. Define the process that determines that validation testing is ready to begin.



Explain how final agreement from all stakeholders is reached. Explain the prerequisites leading up to validation testing.

An Implementation Readiness Review (IRR) is conducted after completion of all validation testing to affirm final agreement from all stakeholders that the <automated system/application> is ready to begin implementation testing. During the IRR, all major findings, resolutions, and related test results from the completed validation testing are assessed. The IRR seeks to ensure that all prerequisites leading up to implementation testing were met.

<Instructions: Describe project specific implementation readiness review prerequisites, review criteria, review processes, communication, or approval processes. Describe who assesses all major findings, resolutions, and test results and what the criteria measures.

An Operational Readiness Review (ORR) is conducted to present all major findings, resolutions, and related test results from all completed validation and implementation testing to all stakeholders and senior leadership to affirm final agreement that the <automated system/application> is ready to move to the Production Environment (PROD) for operational testing. The ORR seeks to ensure that all prerequisites leading up to operational testing were met. Go/No-Go decisions in the production operation of FX projects secure business system owner approval and would be approved by the Technology Standards Committee. Decisions on high risk, high visibility, and public-facing projects may also be escalated to FX Program Governance or FX Executive Governance for approval.

<Instructions: Describe project specific operational readiness review prerequisites, review criteria, review processes, communication, or approval processes.



SECTION 5 PLANNED TESTS

<Instructions: As applicable to the scope of the Test Plan being prepared, describe the various types of testing (test functions) to be performed for the system, application, or situation during the life cycle, taking into consideration the system development methodology that is being employed for the project (e.g., waterfall, prototyping, incremental, spiral, or rapid application development).

Each test function should be described under a separate subsection heading and include a description of the purpose, approach, components, procedures, and techniques that will be used. Also include a statement of the extent of testing to be performed and the rationale for the extent selected, as well as metrics/assessment criteria, for each test function. A separate Test Plan may be prepared to address a specific testing function (e.g., a separate detailed Security Test and Evaluation (ST&E) Test Plan) and referenced in the main Test Plan.

The following is boilerplate text and sub-sections regarding the various test functions that are generally conducted for the majority of projects. This boilerplate text and sub-sections should be considered for inclusion in a Test Plan and may be modified as appropriate for the given project based on the scope of the specific Test Plan being prepared.>

The following are various test functions that will be performed for the <automated system/application>.

Exhibit 5-1: Types of Testing shows test categories, the types of tests performed in each category, and a default recommendation on if the type of test is required.

TEST CATEGORY	TEST TYPE	REQUIRED
Development Testing	Unit Testing is a type of testing where smallest testable parts of an application, called units, are individually and independently tested.	Yes
Integration Testing	Integration Testing is a type of testing to expose defects and validate the system's readiness to meet non-functional requirements.	Yes
System Testing	System Testing is a type of testing to validate programs, functions, or integrated system components meet the stated requirements.	Yes
	Section 508 Testing is a type of testing that allows for an organization to test their digital content's compliance or conformance to the Web Content Accessibility Guidelines (WCAG).	If Interactive Pages



TEST CATEGORY	TEST TYPE	REQUIRED
	<p>Security Testing is a type of testing to validate the solution meets the FX Security Standards.</p>	Yes
	<p>Penetration Testing is a type of testing that simulates cyber-attack on your system to expose vulnerabilities and generates a report on risk exposure.</p>	Yes
Regression Testing	<p>Regression Testing is a type of testing to confirm that a recent program or code change has not adversely affected existing features.</p>	Yes
User Acceptance Testing	<p>User Acceptance Testing is a type of testing performed by the end user to verify/accept the solution before moving to the production environment.</p>	Yes
	<p>Section 508 Testing is a type of testing that allows for an organization to test their digital content's compliance or conformance to the Web Content Accessibility Guidelines (WCAG).</p>	If Interactive Pages
	<p>Security Testing is a type of testing to validate the solution meets the FX Security Standards.</p>	Yes
	<p>Penetration Testing is a type of testing that simulates cyber-attack on your system to expose vulnerabilities and generates a report on risk exposure.</p>	Yes
	<p>Infrastructure Testing is a type of testing that validates infrastructure components against the approved system design.</p>	Yes
	<p>Parallel Testing is a type of testing where two different versions of a solution are tested concurrently with the same input.</p>	Yes
Performance Testing	<p>Performance Testing is to evaluate that the system or component meets the performance requirements for responsiveness, stability, scalability, reliability, speed, and resource usage of the system under varying load.</p>	Yes



TEST CATEGORY	TEST TYPE	REQUIRED
	Load Testing is to evaluate how the system functions under specific loads of users, simultaneously made calls, or processed transactions over a period of time.	Yes
	Infrastructure Testing is a type of testing that validates infrastructure components against the approved system design.	Yes
Build Verification Testing	Build Verification testing/Smoke Test is a type of testing that is done to ensure the deployment to production is stable.	Yes
Operational Readiness Testing	Operation Readiness Testing is a type of testing done to confirm the system/application is ready for turnover to the Operations and Maintenance team.	Yes

Exhibit 5-1: Types of Testing

5.1 ENTRANCE AND EXIT CRITERIA

The following presents the entrance and exit criteria for the various test phases.

Testing Category	Development Testing
Definition	A test that validates individual software components in isolation. Tests should be completed by a team member other than the author of the new code.



<p>Entry Criteria</p>	<ul style="list-style-type: none"> ▪ Technical design documents have been completed and approved by the appropriate stakeholders ▪ Code review has been completed for all impacted modules with each review completed by a team member other than the author of the code ▪ Development Test Plan has been completed and approved ▪ Development Test Cases and Test Scripts have been developed and approved ▪ Development Test Cases and Test Scripts have been loaded into the FX Enterprise Test Management tool ▪ Development Test Data has been identified, created, and is available for testing ▪ Development Test Environment is available and ready for test execution
<p>Exit Criteria</p>	<ul style="list-style-type: none"> ▪ All Development Tests have been executed ▪ Development Test Results documented, reviewed, and signed off with identified stakeholders

Exhibit 5-2: Developmental Testing Entry and Exit Criteria

<p>Testing Category</p>	<p>Integration Testing</p>
<p>Definition</p>	<p>Programs and applications that interoperate are tested together to confirm that they interact according to the requirements.</p>
<p>Entry Criteria</p>	<ul style="list-style-type: none"> ▪ Exit Criteria for Development Test have been reviewed and approved ▪ Integration Test Plan has been completed and approved ▪ Integration Test Scenarios have been developed and approved ▪ Integration Test Cases and Test Scripts have been developed and approved ▪ Integration Test Cases and Test Scripts have been loaded into the FX Enterprise Test Management tool ▪ Integration Test Data has been identified, created, and available for testing ▪ Integration Test Environment is available and ready for test execution
<p>Exit Criteria</p>	<ul style="list-style-type: none"> ▪ All Integration Tests have been executed ▪ All critical and major defects are resolved and validated



	<ul style="list-style-type: none"> ▪ Integration Test Results documented, reviewed, and signed off with identified stakeholders ▪ All defects are closed or deferred by consensus of the Agency, technical teams, and Project Management team
--	---

Exhibit 5-: Integration Testing Entry and Exit Criteria

Testing Category	System Testing
Definition	System testing is a functional test completed once the code is promoted to the System Test environment from the Integration environment. These tests will confirm the code functions according to the design.
Entry Criteria	<ul style="list-style-type: none"> ▪ Exit Criteria for Development and Integration Testing have been reviewed and approved ▪ System Test Plan has been completed and approved ▪ System Test Scenarios have been developed and approved ▪ System Test Cases and Test Scripts have been developed and approved ▪ System Test Cases and Test Scripts have been loaded into the FX Enterprise Test Management tool ▪ System Test Data has been identified, created, and is available for testing ▪ System Test Environment is available and ready for test execution ▪ Appropriate access and training for testers has been completed
Exit Criteria	<ul style="list-style-type: none"> ▪ All System Test Cases have been executed ▪ All critical and major defects are resolved and validated ▪ System Test Results documented, reviewed, and signed off with identified stakeholders ▪ All defects are closed or deferred by consensus of the Agency, technical teams, and Project Management team

Exhibit 5-3: System Testing Entry and Exit Criteria

Testing Category	Regression Testing
Definition	Regression testing is a validation that legacy systems are not disrupted by new code deployment.



<p>Entry Criteria</p>	<ul style="list-style-type: none"> ▪ Exit Criteria for Development, Integration, and System Test have been reviewed and approved ▪ Automated regression test scripts have been loaded into the FX Enterprise Test Management tool ▪ Any manual test scripts required for regression have been documented, reviewed, and entered into the FX Enterprise Test Management tool ▪ Agency and technical team resources are available to execute regression tests ▪ Legacy systems to be tested are available to test
<p>Exit Criteria</p>	<ul style="list-style-type: none"> ▪ All Regression Test Cases have been executed ▪ All critical and major severity defects are resolved and validated ▪ Regression Test Results documented, reviewed, and signed off with identified stakeholders ▪ All defects are closed or deferred by consensus of the Agency, technical teams, and Project Management team

Exhibit 5-4: Regression Testing Entry and Exit Criteria

<p>Testing Category</p>	<p>User Acceptance Testing (UAT)</p>
<p>Definition</p>	<p>The business user tests the impact of the implementation and runs a set of business processes. This is done after the Test team has completed their test case execution and all defects ranked above Minor have been fixed and validated. This test seeks final approval of the system for deployment from the Agency.</p>
<p>Entry Criteria</p>	<ul style="list-style-type: none"> ▪ Exit Criteria for Development, Integration, System, and Regression Test have been reviewed and approved ▪ UAT Test Plan has been completed and approved ▪ UAT Test Scenarios have been developed and approved ▪ UAT Test Cases and Test Scripts have been developed and approved ▪ UAT Test Cases and Test Scripts have been loaded into the FX Enterprise Test Management tool ▪ UAT Test Data has been identified, created, and is available for testing ▪ UAT Test Environment is available and ready for test execution ▪ Appropriate access and training for testers has been completed



Exit Criteria	<ul style="list-style-type: none"> ▪ All UAT Test Cases have been executed ▪ All critical and major defects are resolved and validated ▪ UAT Test Results documented, reviewed, and signed off with identified stakeholders ▪ All defects are closed or deferred by consensus of the Agency, technical teams, and Project Management team
----------------------	---

Exhibit 5-5: User Acceptance Testing Entry and Exit Criteria

Test Category	Performance Testing
Definition	<p>Performance testing focuses on the following two key areas:</p> <ul style="list-style-type: none"> ▪ Performance Testing: This is to evaluate that the system or component meets the performance requirements for responsiveness, stability, scalability, reliability, speed, and resource usage of the system under varying load. ▪ Load Testing: This is to evaluate how the system functions under specific loads of users, simultaneously made calls, or processed transactions over a period of time.
Entry Criteria	<ul style="list-style-type: none"> ▪ Non-Functional Requirements, based on the performance expectations, have been documented and approved ▪ Exit Criteria for Development, Integration, System, Regression, and User Acceptance Test have been reviewed and approved ▪ Performance Test Plan has been completed and approved ▪ Performance Test Scenarios have been developed and approved ▪ Performance Test Cases and Test Scripts have been developed and approved ▪ Performance Test Cases and Test Scripts have been loaded into the FX Enterprise Test Management tool ▪ Performance Test Data has been identified, created, and is available for testing ▪ Performance Test Environment is available and ready for test execution ▪ Appropriate access and training for testers has been completed
Exit Criteria	<ul style="list-style-type: none"> ▪ All Performance Test Cases have been executed ▪ All critical and major defects are resolved and validated ▪ Performance Test Results documented, reviewed, and signed off with identified stakeholders



	<ul style="list-style-type: none"> All defects are closed or deferred by consensus of the Agency, technical teams, and Project Management team
--	---

Exhibit 5-: Performance Testing Entry and Exit Criteria

Testing Category	Build Verification Testing / Smoke Testing
Definition	Build Verification Testing is a subset of test cases that cover the most important functionality of a component or system, used to aid the assessment of whether main functions of the software appear to work correctly after production deployment.
Entry Criteria	<ul style="list-style-type: none"> Exit Criteria for Development, Integration, System, Regression, User Acceptance Testing, and Performance Testing have been reviewed and approved Production Deployment is completed Build Verification Test Plan has been completed and approved Build Verification Test Scenarios have been developed and approved. Only critical scenarios should be included Build Verification Test Cases and Test Scripts have been developed and approved Build Verification Test Cases and Test Scripts have been loaded into the FX Enterprise Test Management tool Build Verification Test Data has been identified, created, and is available for testing Build Verification Test Environment is available and ready for test execution Appropriate access and training for testers has been completed
Exit Criteria	<ul style="list-style-type: none"> All Build Verification Test Cases have been executed All critical and major defects are resolved and validated Build Verification Test Results documented, reviewed, and signed off with identified stakeholders All defects are closed or deferred by consensus of the Agency, technical teams, and Project Management team

Exhibit 5-6: Build Verification Testing Entry and Exit Criteria

Testing Category	Operational Readiness Testing
Definition	Operational Readiness Testing is done to confirm the system/application is ready for turnover to the Operations and Maintenance team.



Entry Criteria	<ul style="list-style-type: none"> ▪ Operational Readiness Test plan is completed and approved ▪ Development, Integration, System, Regression, User Acceptance, Performance, and Build Verification Testing are complete ▪ Roll-Back plan has been reviewed and approved
Exit Criteria	<ul style="list-style-type: none"> ▪ Turnover steps have been executed successfully and all involved have been made aware of their required activities and the schedule ▪ Testing on legacy systems, if required, is complete ▪ Testing completed on sufficiently large sample size to provide confidence in readiness ▪ Roll-Back plan has successfully passed testing

Exhibit 5-: Operational Readiness Testing Entry and Exit Criteria

5.2 DEVELOPMENT TESTING

The following list presents the development test functions the team will perform for the <automated system/application>:

<Development test function>

The results from development testing will be included in the Version Description Document (VDD) for the specific system build or release that is being transitioned into subsequent testing phases.

5.3 INTEGRATION TESTING

The following list presents the integration test functions the team will perform for the <automated system/application>:

<Integration test function>

The results from integration testing will be documented in one or more corresponding Test Summary Reports (TSRs).

5.4 SYSTEM TESTING

The following list presents the system test functions the team will perform for the <automated system/application>:

<System test function>



The results from system testing will be documented in one or more corresponding TSRs.

5.5 REGRESSION TESTING

The following list presents the regression test functions the team will perform for the <automated system/application>:

<Regression test function>

The results from regression testing will be documented in one or more corresponding TSRs.

5.6 USER ACCEPTANCE TESTING

The following list presents the user acceptance test functions the team will perform for the <automated system/application>:

<User Acceptance test function>

The results from user acceptance testing will be documented in one or more corresponding TSRs.

5.7 PERFORMANCE TESTING

The following list presents the performance test functions the team will perform for the <automated system/application>:

<Performance test function>

The results from performance testing will be documented in one or more corresponding TSRs.

5.8 BUILD VERIFICATION TESTING / SMOKE TESTING

The following list presents the build verification test functions the team will perform for the <automated system/application>:

<Build Verification test function>

The results from build verification testing will be documented in one or more corresponding TSRs.

5.9 OPERATIONAL READINESS TESTING

The following list presents the operational readiness test functions the team will perform for the <automated system/application>:



<Operational Readiness test function>

The results from operational readiness testing will be documented in one or more corresponding TSRs.



SECTION 6 TEST PROGRESSION

<Instructions: As applicable to this Test Plan, explain the planned sequence or progression of the prescribed tests. Identify existing dependencies that affect the conduct and progression of test activities. Also identify any regularly held meetings or reports that provide information on or that may affect testing (e.g., Change Control Board (CCB) meetings, status reports, etc.).>



SECTION 7 FEATURE TESTING

7.1 FEATURES TO BE TESTED

<Instructions: List and describe system functions/features that are to be tested. The table below provides an example that identifies features, software items, criticality to deployment success, testing priority, and notes. An explanation should be provided for the methodology used to define the criticality to deployment success and testing priority levels established for the given project.>

FEATURE	SOFTWARE	CRITICALITY TO DEPLOYMENT SUCCESS	TESTING PRIORITY	NOTES
<Feature>	<Software>	<Criticality to deployment success>	<Period of usage>	<Notes>

Exhibit 7-1: Features to Be Tested

7.2 FEATURES NOT TO BE TESTED

<Instructions: List and describe the system functions/features not planned to be tested and explain why. The table below provides an example that identifies features, software items, criticality to deployment success, testing priority, and notes. Generally, only identify non-critical functions/features with a designated low priority as features not to be tested due to resource or other identified constraints.>

FEATURE	SOFTWARE	CRITICALITY TO DEPLOYMENT SUCCESS	TESTING PRIORITY	NOTES
<Feature>	<Software>	<Criticality to deployment success>	<Period of usage>	<Notes>

Exhibit 7-2: Features Not to Be Tested



SECTION 8 TEST CASES

<Instructions: Describe the measures taken to document and prioritize test cases, the controls applied to them, and how/where they are stored (e.g., FX Projects Repository, Application Life Cycle Management Tool, CD, library, etc.). Generally, test cases are traditionally documented in a separate Test Case Specification that should be referenced within this section.>

SECTION 9 DEFECT TRACKING AND REPORTING

<Instructions: Describe the measures taken to document and track identified defects; the taxonomy used to classify the impact, type, and severity of the defect; the resolutions applied and subsequent test results; how/where they are stored (e.g., FX Projects Repository, Application Life Cycle Management Tool, CD, library, etc.); and the methods, schedule, and audiences to whom they are reported. >

9.1 DEFECT MANAGEMENT

Defect Management is the process of detecting and fixing bugs that occur in the software. During the Test Execution phases of a project, Test Cases can either meet the expected results or not. When expected results are not met, a defect has been detected and when logged, initiates the Defect Life Cycle. The Defect Life Cycle can be further described as:

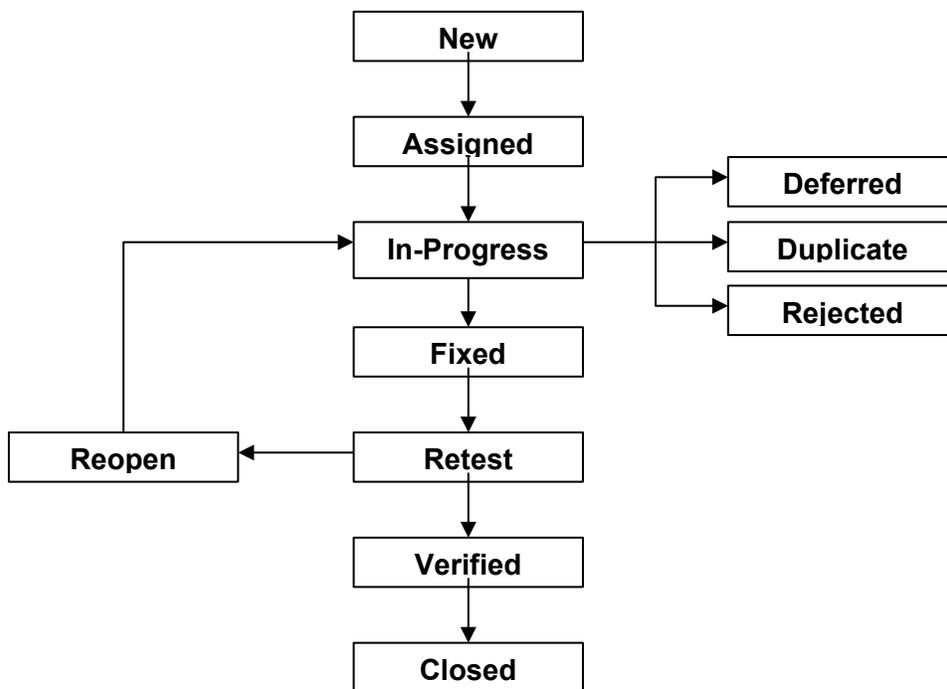


Exhibit 9-1: Defect Life Cycle

- **New:** A new defect is logged and posted for the first time. The initial status is “New.”
- **Assigned:** Once the defect is assigned to the development team, that status is updated to “Assigned.”
- **In-Progress:** The developer starts analyzing and working on the defect fix. The status is updated to “In-Progress.”
- **Fixed:** When a developer makes the necessary code change and verifies the change, he or she updates the defect status to “Fixed.”
- **Retest:** Once the defect is fixed the developer loads the code into the appropriate Test Environment and the status is updated to “Retest.”
- **Verified:** The tester executes the test case and verifies the results. If expected results match the actual results, the status is updated to “Verified.”
- **Closed:** If the defect is “Verified” and has no other dependencies, the status is updated to “Closed.”
- **Reopen:** If the defect persists, the tester changes the status to “Reopen.”
- **Deferred:** If the defect is not a priority and if it is expected to get fixed in the next release, then the status is updated to “Deferred.”
- **Duplicate:** If the defect is found to be a duplicate, the status is updated to “Duplicate” and a cross reference to matching defect is noted.
- **Rejected:** If it is determined that the defect is not a genuine defect, then the status can be updated to “Rejected.” The tester can then reevaluate, and if necessary, clarify and set the status to “Reopen.”

9.2 SEVERITY AND PRIORITY

Severity and Priority levels are assigned to each individual defect. Priority is associated with scheduling and severity is associated with business standards. The following presents levels and descriptions to be used:

9.2.1 SEVERITY CATEGORIES

Severity defines the degree of impact. It reflects the degree or intensity of a specific defect adversely impacting the software. A defect may be further categorized into the following:

- **Critical:** The defects categorized as 'critical' need immediate attention and treatment. A critical defect directly affects the critical and essential functionalities, which affect the software product or its functionality on a large scale, such as failure of a feature/functionality, system crash, system down, etc. It does not have a workaround.
- **Major:** Defects are responsible for affecting the core and major functionalities of a software product. Although, these defects do not result in complete system failure it may adversely affect several major functions. It has a workaround, but it is not obvious, and is difficult or complex to perform.
- **Minor:** These defects do not have any significant influence on the software product. The results of these defects may be seen as the product or function is not working; however, it does not stop



the user's ability to execute tasks or continue by using some other alternative. It has an uncomplicated workaround.

- **Cosmetic:** These types of defects have no impact on the working of the product and sometimes they are ignored and skipped, such as spelling or grammatical mistakes.

9.2.2 PRIORITY CATEGORIES

Priority defines the desired order of fixing the defects. Priorities may be further categorized into the following:

- **High:** The defect must be resolved as soon as possible because of a technical or business necessity.
- **Medium:** The defect should be resolved in the normal course of development activities. It can wait until a new build or version is created.
- **Low:** The defect is an irritant which should be repaired, but repair can be deferred until after more serious defects have been fixed.



SECTION 10 TEST ENVIRONMENT

<Instructions: Provide details and a graphical presentation of the environmental components required to test the system to include hardware, software, communications, and any other resources used to configure the test environment(s), as well as any security considerations. If multiple test sites will be used, each test site should be explicitly identified and the test environment for each test site appropriately described. If multiple test sites use the same or similar test environments, they may be discussed together with the differences clearly identified. The test environment(s) should reflect the planned production environment as closely as possible. Also provide details where the test environment(s) does not mirror the production environment.>

10.1 TESTING ENVIRONMENTS

Below are the minimum number of environments that are required for each FX project to support the solution. The project must adhere to the naming standards for these environments. These environments must be flexible to integrate with other solutions, with built-in burst capacity to handle burst periods or potential growth. The environments must be implemented with procedures, technologies, and capabilities that support non-disruptive configuration changes.

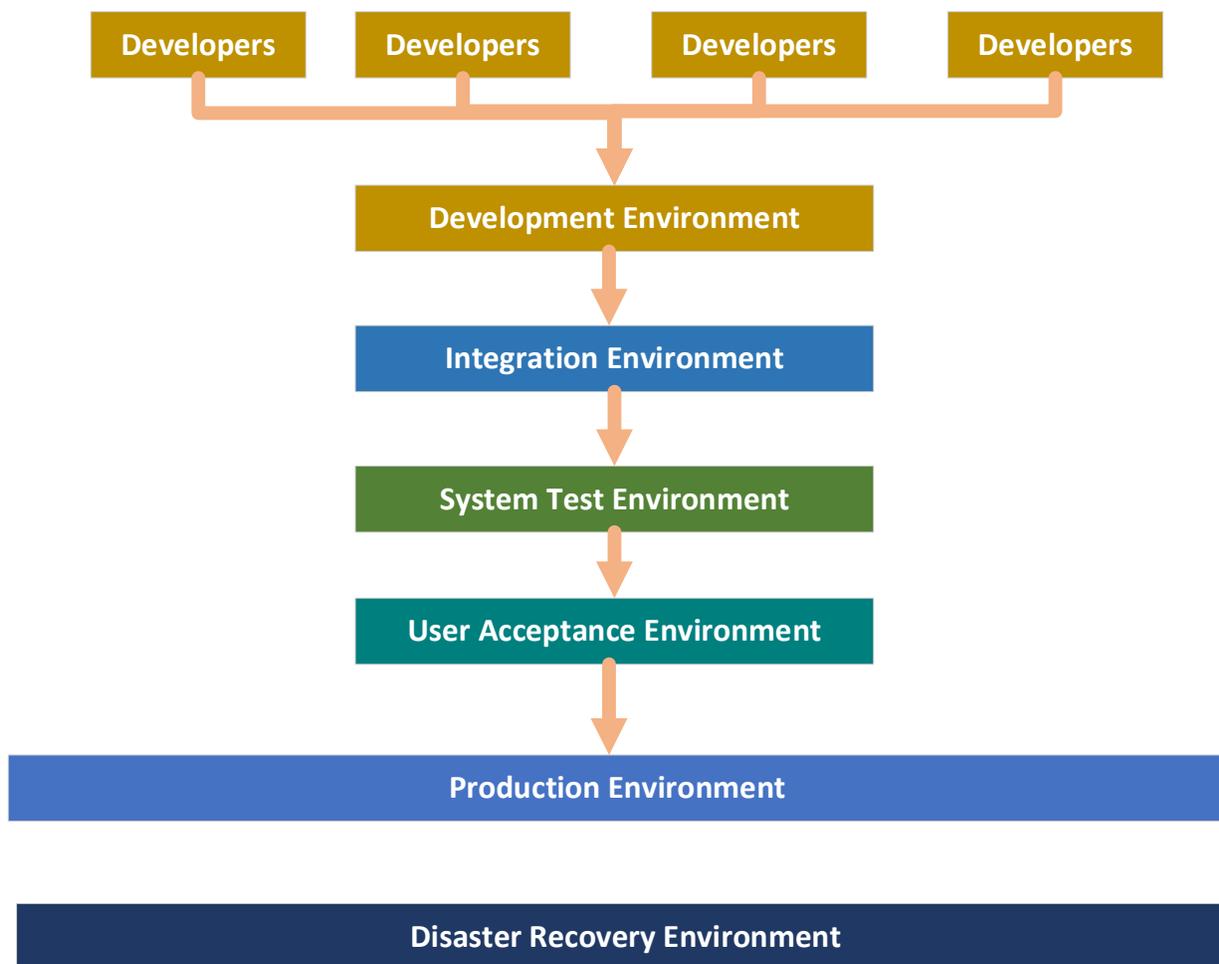


Exhibit 10-1: FX Environments Diagram



ENVIRONMENTS	DESCRIPTION
Development	<ul style="list-style-type: none"> ▪ Working environment for all developers ▪ Developers check-in and check-out their code ▪ Unit testing is performed in this environment
Integration	<ul style="list-style-type: none"> ▪ Work from the entire project team is combined and validated ▪ Integration testing is done to expose defects and validate the system's readiness to meet non-functional requirements
System Test	<ul style="list-style-type: none"> ▪ Builds from Integration environment are promoted to test the system from end-to-end for functional and non-functional components ▪ System testing is done to validate a program, function, or integrated system component meets its requirements ▪ Regression Testing is done following modifications to the system to verify defects have not been introduced or uncovered from changes made
User Acceptance Test	<ul style="list-style-type: none"> ▪ Identical to the production environment in configuration ▪ Used to simulate the production environment to validate functional requirements ▪ Can be used as a training environment ▪ User Acceptance Testing conducted by the Agency to determine whether a system satisfies the defined user acceptance criteria in an isolated environment
Production	<ul style="list-style-type: none"> ▪ Operational Environment with the latest released version of the system, accessible to the client/end users ▪ Build Verification testing/Smoke Test is done to ensure the deployment to production is stable ▪ Can be used for Performance Testing that verifies the performance of a system will meet the Performance Standards and measures the behavior of the system with increasing load ▪ Operation Readiness Testing is done to confirm the system/application is ready for turnover to the Operations and Maintenance team
Disaster Recovery	<ul style="list-style-type: none"> ▪ Timely failover for business continuation services in the event of a disaster ▪ Disaster Recovery testing is done annually to validate disaster recovery preparedness

Exhibit 10-2: FX Environments Description

ENVIRONMENTS	TESTING CATEGORY
Development	Development testing
Integration	Integration Testing



ENVIRONMENTS	TESTING CATEGORY
System Test	System Testing Regression Testing
User Acceptance Test	User Acceptance Testing
Production	Operational Readiness Testing Build Verification Testing/Smoke Test Performance Testing

Exhibit 10-3: FX Environment - Testing Crosswalk

10.2 HARDWARE

<Instructions: Identify by name, number, and version, as applicable, all computer hardware, interfacing equipment, communications equipment, peripherals, etc., that will be required at each test site. Describe the purpose of each item and state the period of usage and the number of each item needed.>

HARDWARE ITEM	PURPOSE	PERIOD OF USAGE	NUMBER NEEDED	ISSUES
<Hardware item>	<Purpose>	<Period of usage>	<Number needed>	<Issues>

Exhibit 10-4: Testing Hardware

10.3 SOFTWARE

<Instructions: Identify by name, number, and version, as applicable, all software items (e.g., operating systems, compilers, communications software, related applications software, databases, input files, code auditors, dynamic path analyzers, test drivers, pre-processors, test data generators, test control software, other special test software, post-processors, etc.) that will be required at each test site. Describe the purpose of each item, its media, and state the period of usage and the number of each item needed. Also identify the proprietary nature and any licensing issues associated with each item.>

SOFTWARE ITEM	PURPOSE	MEDIA	PERIOD OF USAGE	NUMBER NEEDED	ISSUES
<Software item>	<Purpose>	<Media>	<Period of usage>	<Number needed>	<Issues>

Exhibit 10-5: Testing Software



10.4 TEST DATA

<Instructions: Provide a detailed description of the test data to be used for the various testing activities. If real, personally identifiable information (PII) will be needed, a separate Data Use Agreement (DUA) must also be completed. If PII will be used, identify how this test data will be protected and controlled.>

10.5 OTHER MATERIALS

<Instructions: Identify and describe any other materials needed for the testing at the test site(s). These materials may include manuals or other forms of instruction. Identify the type and quantity of the materials, as applicable.>

10.6 INSTALLATION, TESTING, AND CONTROL

<Instructions: Identify plans for establishing the test environment at each test site and test each element prior to its use. Also describe how the test environment will be controlled and maintained. The schedule for establishing the test environment should be documented in the Project Schedule and referenced here as appropriate.>

10.7 SECURITY

<Instructions: Identify any security or privacy issues associated with the test environment, including any issues regarding PII not previously addressed in the Test Data section.>



SECTION 11 TEST DELIVERABLES

<Instructions: Briefly describe the documents that are to be produced in support of and/or as output of the testing effort. For example, a Test Case-to-Requirements Traceability Matrix should be included as part of a separate Test Case Specification document and referenced here. Other documents may include, but are not limited to, Test Incident Reports (TIRs) and Test Summary Reports (TSRs).>

Future iterations of this document are to define testing documentation consistency across projects, sample test plans, requirements for test plans documenting content, methodology, prioritization, and progression of testing activities.>

11.1 TEST CASE-TO-REQUIREMENTS TRACEABILITY MATRIX

A Test Case-to-Requirements Traceability Matrix that maps all requirements contained within the Requirements Document to their corresponding test cases, will be prepared and included in a separate Test Case Specification document.

11.2 TEST INCIDENT REPORTS

Test Incident Reports will be used during the test process to identify, capture, track, and resolve unexpected results, problems, or defects identified during testing.

11.3 TEST SUMMARY REPORT

A TSR will be prepared at the end of all testing to document the results of all tests this Test Plan (TP) prescribes.



SECTION 12 TEST SCHEDULE AND MILESTONES

<Instructions: List the milestone events and dates for all testing activities, including each test site as appropriate.>

TESTING MILESTONE	DATE
<Testing Milestone>	<Date>

Exhibit 12-1: Testing Milestones



SECTION 13 TEST ROLES AND RESPONSIBILITIES

<Instructions: Identify the number, type, and skill level of the personnel that will be needed from each organization to participate in the testing activities, during each of the prescribed tests, and at the designated test site(s); describe the roles and responsibilities of each. Include the names of the individuals, if known.>

SKILL TYPE	SKILL LEVEL	# PERSONNEL	ASSIGNED STAFF	LENGTH OF TIME NEEDED	ROLE
<Skill type>	<Skill level>	<#>	<First name last name>	<Length of time needed>	<Role>

Exhibit 13-1: Testing Resources

13.1 ORIENTATION PLAN

<Instructions: Describe any orientation and training to be given before and/or during the testing. This training may include user instruction, operator instruction, maintenance and control group instruction, and orientation briefings to test team personnel. If extensive training is anticipated, a separate Training Plan and Training Artifacts may be developed and referenced here.>



APPENDICES