

AHCA Florida Health Care Connections (FX)

<<Insert Project Name Here>>

Testing Management Plan

Version: 001

Date: Month Day, YYYY

Author: [Author]

Submitted To: AHCA FX Program Administration Team





Revision History

DATE	VERSION	DESCRIPTION	AUTHOR
M/D/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
M/D/YYYY		



Table of Contents

Section 1	Introduction	1
1.1	Background	1
1.2	Purpose	1
1.3	Scope Statement.....	1
1.4	Goals and Objectives	2
1.5	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	8
5.2	Development Testing Phase.....	13
5.3	Integration Testing Phase	13
5.4	System Testing Phase	14
5.5	Regression Testing Phase.....	14
5.6	User Acceptance Testing Phase	14
5.7	Performance Testing Phase	14
5.8	Build Verification Testing / Smoke Testing Phase.....	14
5.9	Operational Readiness Testing Phase	15
Section 6	Test Progression.....	16
Section 7	Feature Testing.....	17
7.1	Features To Be Tested	17
7.2	Features Not To Be Tested.....	17



Section 8	Test Cases	18
Section 9	Defect Tracking and Reporting	19
9.1	Defect Management	19
9.2	Severity and Priority	20
9.2.1	Severity Categories	20
9.2.2	Priority Categories	21
Section 10	Test Environment	22
10.1	Testing Environments	22
10.2	Hardware	24
10.3	Software	25
10.4	Test Data	25
10.5	Other Materials	25
10.6	Installation, Testing, and Control	25
10.7	Security	26
Section 11	Test Deliverables	27
11.1	Test Case-to-Requirements Traceability Matrix	27
11.2	Defect Reports	27
11.3	Test Summary Report	27
Section 12	Test Milestones	28
Section 13	Orientation	29
13.1	Orientation Plan	29
Appendices	30



Table of Exhibits

Exhibit 2-1: Roles and Responsibilities	3
Exhibit 5-1: Types of Testing	8
Exhibit 5-2: Developmental Testing Entry and Exit Criteria	9
Exhibit 5-3: Integration Testing Entry and Exit Criteria	10
Exhibit 5-4: System Testing Entry and Exit Criteria	10
Exhibit 5-5: Regression Testing Entry and Exit Criteria	11
Exhibit 5-6: User Acceptance Testing Entry and Exit Criteria.....	11
Exhibit 5-7: Performance Testing Entry and Exit Criteria	12
Exhibit 5-8: Build Verification Testing Entry and Exit Criteria	13
Exhibit 5-9: Operational Readiness Testing Entry and Exit Criteria	13
Exhibit 7-1: Features to Be Tested	17
Exhibit 7-2: Features Not to Be Tested	17
Exhibit 9-1: Defect Life Cycle	19
Exhibit 10-1: FX Environments Diagram	23
Exhibit 10-2: FX Environments Description.....	24
Exhibit 10-3: FX Environment - Testing Crosswalk.....	24
Exhibit 10-4: Testing Hardware	25
Exhibit 10-5: Testing Software.....	25
Exhibit 12-1: Testing Milestones.....	28



SECTION 1 INTRODUCTION

1.1 BACKGROUND

The Florida Agency for Health Care Administration (AHCA or Agency) is adapting to the changing landscape of healthcare 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 is complex; it includes services, business processes, data management and processes, technical processes within the Agency, and interconnections and touchpoints with systems necessary for administration of the Florida Medicaid program that reside outside the Agency. The future of the Florida Medicaid Enterprise integration is to allow the Agency to secure services that can interoperate and communicate without relying on a common platform or technology.

The Florida Medicaid Management Information System (FMMIS) has historically been the central system within the Florida Medicaid Enterprise; functioning as the single, integrated system for claims processing and information retrieval. As the Medicaid program has grown more complex, the systems needed to support the Florida Medicaid Enterprise have grown in number and complexity.

The Medicaid Enterprise System (MES) Procurement Project was re-named Florida Health Care Connections (FX) in the summer of 2018. FX is a multi-year transformation to modernize the current Medicaid technology using a modular approach, while simultaneously improving overall Agency functionality and building better connections to other data sources and programs.

1.2 PURPOSE

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

<The template is designed as a guide to produce a deliverable that is complete, requires minimal iterations of review, and acknowledges an understanding of project standards. The gray text throughout the document provides context for the section and should be removed from the deliverable. The black text is boilerplate language that should stay in the deliverable and any modifications are to be documented as to the reason.>

1.3 SCOPE STATEMENT

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

- Roles and Responsibilities



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

1.4 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>

1.5 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.

- CMS Target Life Cycle (TLC) 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

The project risk log on the FX Projects Repository is the appropriate place for creating and maintaining all risks. Any risk or issue impacting the schedule shall be highlighted in the status report. All issues or risks that require escalation shall follow the process outlined in the *P-2: FX Project Management Standards*.



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's 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

Readiness reviews are defined in this section, and the details of the reviews are captured in other project artifacts. The entrance and exit criteria defined in this document affirm the testing phase transitions.

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

An Implementation Readiness Review (IRR) is conducted after completion of all validation testing to affirm final agreement from all stakeholders that the 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.



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 application is ready to move to the Production (PROD) environment 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 the FX Implementation Team (FXIT) or the FX Executive Steering Committee (ESC) for approval.



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 about 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 subsections regarding the various test functions that are generally conducted for most projects. This boilerplate text and subsections 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.>

Exhibit 5-1: Types of Testing shows the types of tests performed and a default recommendation whether test is required.

TEST TYPE	REQUIRED
Unit Testing is a type of testing where smallest testable parts of an application, called units, are individually and independently tested.	Yes
Integration Testing is a type of testing to expose defects and validate the system's readiness to meet non-functional requirements.	Yes
System Testing is a type of testing to validate that 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
Security Testing is a type of testing to validate that the solution meets the FX Security Standards.	Yes
Penetration Testing is a type of testing that simulates a cyber-attack on your system to expose vulnerabilities and generates a report on risk exposure.	Yes



TEST TYPE	REQUIRED
Vulnerability Testing is a type of testing that detects the four different types of vulnerabilities: human-social, physical, economic, and environmental.	Yes
Regression Testing is a type of testing to confirm that a recent program or code change has not adversely affected existing features.	Yes
User Acceptance Testing is a type of testing performed by the end user to verify/accept the solution before moving to the Production environment.	Yes
Infrastructure Testing is a type of testing that validates infrastructure components against the approved system design.	Yes
Parallel Testing is a type of testing where two different versions of a solution are tested concurrently with the same input.	Yes, if there is an existing solution
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.	Yes
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
Build Verification Testing/Smoke Test is a type of testing that is done to ensure the deployment to production is stable.	Yes
Disaster Recovery Testing is a type of testing that is done to confirm the disaster recovery plan works as designed.	Yes
Operational Readiness Testing is a type of testing done to confirm that 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.

Exhibit 5-2: Developmental Testing Entry and Exit Criteria presents the entrance and exit criteria for Developmental Testing



<Review and approval of the entrance and exit criteria must follow FX documentation standards, contractual obligations, or any other defined process. The review and approval strategy should be described in this section.

The prior testing required to meet entry criteria may vary depending on specific testing challenges, in which case, the prior testing should be updated to reflect the actual plan. Exceptions to the standard entrance criteria or exit criteria must be documented and approved by the Agency. Test phases should be added or removed to meet the needs of the project.>

TESTING PHASE	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.
Entry Criteria	<ul style="list-style-type: none"> ▪ Applicable 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
Exit Criteria	<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

Exhibit 5-3: Integration Testing Entry and Exit Criteria presents the entrance and exit criteria for the Integration Testing.

TESTING PHASE	INTEGRATION TESTING
Definition	Programs and applications that interoperate are tested together to confirm that they interact according to the requirements.
Entry Criteria	<ul style="list-style-type: none"> ▪ Exit Criteria for Development Testing 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
Exit Criteria	<ul style="list-style-type: none"> ▪ All Integration Tests have been executed ▪ All critical and major defects are resolved and validated ▪ Integration Test Results documented, reviewed, and signed off with identified stakeholders



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

Exhibit 5-3: Integration Testing Entry and Exit Criteria

Exhibit 5-4: System Testing Entry and Exit Criteria presents the entrance and exit criteria for the System Testing.

TESTING PHASE	SYSTEM TESTING
Definition	System testing is a functional test completed once the code is promoted to the designated System Testing environment from a lower environment. These tests will confirm that 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 has been granted and confirmed 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 5-4: System Testing Entry and Exit Criteria

Exhibit 5-5: Regression Testing Entry and Exit Criteria presents the entrance and exit criteria for the Regression Testing.

TESTING PHASE	REGRESSION TESTING
Definition	Regression testing is a validation that legacy systems are not disrupted by new code deployment.
Entry Criteria	<ul style="list-style-type: none"> Exit Criteria for Development, Integration, and System Testing 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



TESTING PHASE	REGRESSION TESTING
Exit Criteria	<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 5-5: Regression Testing Entry and Exit Criteria

Exhibit 5-6: User Acceptance Testing Entry and Exit Criteria presents the entrance and exit criteria for the User Acceptance Testing.

TESTING PHASE	USER ACCEPTANCE TESTING (UAT)
Definition	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.
Entry Criteria	<ul style="list-style-type: none"> Exit Criteria for Development, Integration, System, and Regression Testing 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 has been granted and confirmed 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 5-6: User Acceptance Testing Entry and Exit Criteria

Exhibit 5-7: Performance Testing Entry and Exit Criteria presents the entrance and exit criteria for the Performance Testing.

TEST PHASE	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



TEST PHASE	PERFORMANCE TESTING
Entry Criteria	<ul style="list-style-type: none"> ▪ 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 ▪ Non-Functional Requirements, based on the performance expectations, have been documented and approved ▪ Exit Criteria for Development, Integration, System, Regression, and User Acceptance Testing 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 has been granted and confirmed ▪ 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 ▪ All defects are closed or deferred by consensus of the Agency, technical teams, and Project Management team

Exhibit 5-7: Performance Testing Entry and Exit Criteria

Exhibit 5-8: Build Verification Testing Entry and Exit Criteria presents the entrance and exit criteria for the Build Verification Testing.

TESTING PHASE	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 has been granted and confirmed ▪ Training for testers has been completed



TESTING PHASE	BUILD VERIFICATION TESTING / SMOKE TESTING
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 5-8: Build Verification Testing Entry and Exit Criteria

Exhibit 5-9: Operational Readiness Testing Entry and Exit Criteria presents the entrance and exit criteria for the Operational Readiness Testing.

TESTING PHASE	OPERATIONAL READINESS TESTING
Definition	Operational Readiness Testing is done to confirm that the system/application is ready for turnover to the Operations and Maintenance team.
Entry Criteria	<ul style="list-style-type: none"> Operational Readiness Testing plan is completed and approved Development, Integration, System, Regression, User Acceptance, Performance, and Build Verification Testing are complete Rollback 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 Rollback plan has successfully passed testing

Exhibit 5-9: Operational Readiness Testing Entry and Exit Criteria

5.2 DEVELOPMENT TESTING PHASE

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

<Development test function>

5.3 INTEGRATION TESTING PHASE

The following list presents the integration test functions that 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 PHASE

The following list presents the system test functions that 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 PHASE

The following list presents the regression test functions that 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 PHASE

The following list presents the user acceptance test functions that 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 PHASE

The following list presents the performance test functions that 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 PHASE

The following list presents the build verification test functions that 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 PHASE

The following list presents the operational readiness test functions that 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. The details of parallel testing should be included in this section. 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

Exhibit 7-1: Features to Be Tested describes system functions/features that are to be tested.

<Instructions: List and describe system functions/features that are to be tested. The table below provides an example that identifies features, description, and notes.>

FEATURE/FUNCTION	DESCRIPTION	NOTES
<Feature>	<Software>	<Notes>

Exhibit 7-1: Features to Be Tested

7.2 FEATURES NOT TO BE TESTED

Exhibit 7-2: Features Not to Be Tested describe the system functions/features not planned to be tested.

<Instructions: List and describe the system functions/features not planned to be tested and explain why.>

FEATURE/FUNCTION	DESCRIPTION	NOTES/EXPLANATION
<Feature>	<Software>	<Notes>

Exhibit 7-2: Features Not to Be Tested



SECTION 8 TEST CASES

<Instructions: Test cases and test scripts shall be documented, executed, and maintained in the FX test repository, Zephyr Scale. Zephyr Scale has the functionality to export test case data if needed.>

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. **Exhibit 9-1: Defect Life Cycle** displays the process flow of the Defect Life Cycle. The Defect Life Cycle can be further described as:

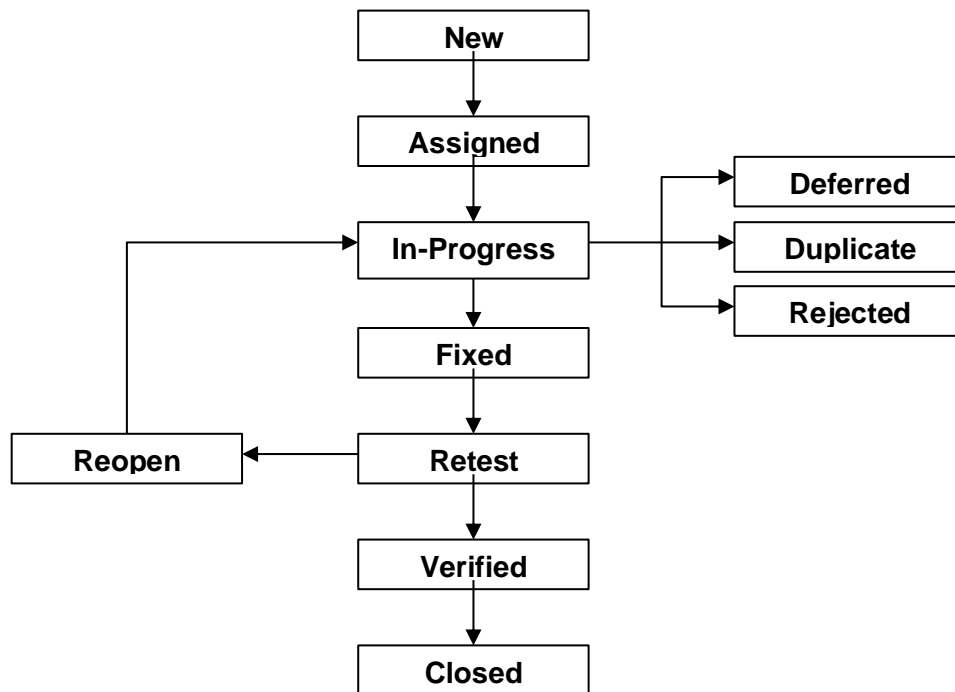


Exhibit 9-1: Defect Life Cycle

- **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 subsections presents the 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. Defect does not have a work-around.
- **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. Defect has a work-around, 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. Defect has an uncomplicated work-around.



- **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

Exhibit 10-1: FX Environments Diagram and **Exhibit 10-2: FX Environments Description** display 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.

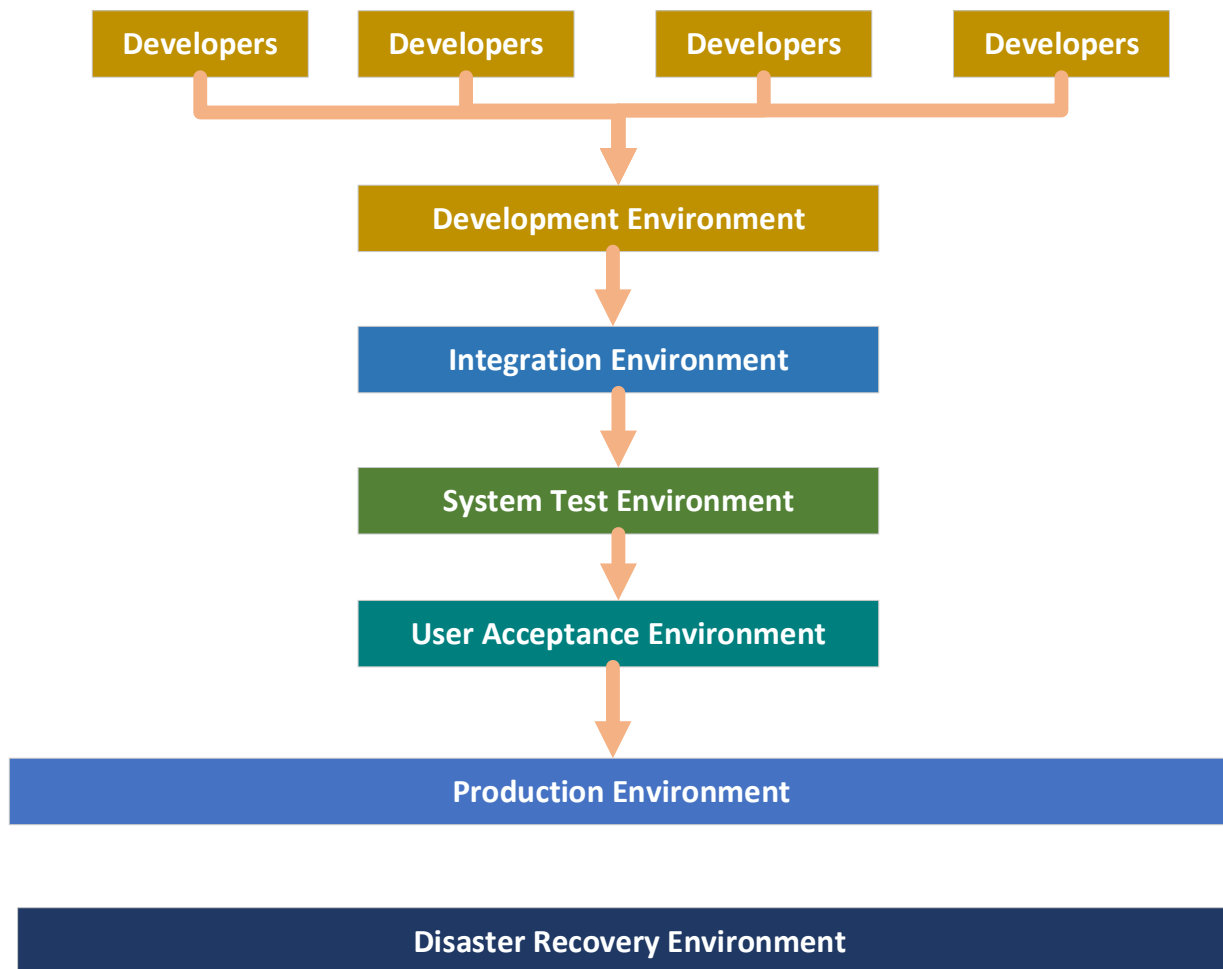


Exhibit 10-1: FX Environments Diagram

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 the Integration environment are promoted to test the system from end-to-end for functional and non-functional components System Testing is done to validate that a program, function, or integrated system component meets its requirements Regression Testing is done following modifications to the system to verify that defects have not been introduced or uncovered from changes made



ENVIRONMENTS	DESCRIPTION
User Acceptance	<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

Exhibit 10-3: FX Environment - Testing Crosswalk lists the testing types that will be executed in each environment.

<Instructions: Add the testing types that will be executed in each environment. The association of testing type to environment can vary depending on project needs.>

ENVIRONMENTS	TESTING TYPE
Development	<ul style="list-style-type: none"> Unit Testing Security
Integration	<ul style="list-style-type: none"> Integration Testing Section 508 Testing Infrastructure
System Test	<ul style="list-style-type: none"> System Testing Regression Testing
User Acceptance	<ul style="list-style-type: none"> User Acceptance Testing Load Testing Performance Testing
Disaster Recovery	<ul style="list-style-type: none"> Disaster Recovery Testing
Production	<ul style="list-style-type: none"> Operational Readiness Testing Build Verification Testing/Smoke Test Penetration Vulnerability Parallel

Exhibit 10-3: FX Environment - Testing Crosswalk

10.2 HARDWARE

Exhibit 10-4: Testing Hardware identifies special devices required for testing or to be tested.



<Instructions: Identify any special devices required for testing or to be tested.>

HARDWARE ITEM	PURPOSE
<Hardware item>	<Purpose>

Exhibit 10-4: Testing Hardware

10.3 SOFTWARE

Exhibit 10-5: Testing Software identifies any specific software needed to execute test cases as well as the software that is being tested.

<Instructions: Identify any specific software needed to execute test cases as well as the software that is being tested (e.g., test data generators, test control software, other special test software, etc.). Yes or no should be added to the Agency Given Access column to indicate if the Agency will have access to the software. Yes or no should be added to the Agency Training column to indicate if the vendor will train Agency staff.>

SOFTWARE ITEM	PURPOSE	AGENCY GIVEN ACCESS	AGENCY TRAINING
<Software item>	<Purpose>	<Yes/No>	<Yes/No>

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: Describe how the test environment will be controlled and maintained. The schedule for establishing the test environment should be documented in the FX Project's 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. This section does not pertain to executing security test cases.>



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 document and referenced here. Other documents may include, but are not limited to, Defect Reports and 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 artifact.

11.2 DEFECT REPORTS

Defect 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 MILESTONES

Exhibit 12-1: Testing Milestones lists the testing milestone events.

<Instructions: List the milestone events. All scheduled testing activities are maintained in the Integrated Master Schedule.>

TESTING MILESTONE

<Testing Milestone>

Exhibit 12-1: Testing Milestones



SECTION 13 ORIENTATION

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