|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Identification** | | | | |
| Initiating Organization: | XXXXXXX | Date: | ---------- | |
| Project Name: | -------- 15.1 Release | Project Code: | XXXX | |
| Application Name: | -------- | Iteration: | N/A | |
| Release Number: | 15.1 | Change Type: | Enhancement | |
| Methodology: | XXXX | | | |
|  |  |  |  | |
| Project Sponsor: | XXX XXXX | | |  |
| Project Manager: | XXX XXXX | | |  |
| Approver #1: | XXX XXXX | | |  |
| Approver #2: | XXX XXXX | | |  |
|  |  |  |  | |

**Document Change Control Log**

This log is updated each time this document is updated. The log identifies the version number, the date the revisions were completed, a brief description of the changes, and the author.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version #** | **Date** | **Sections / Pages Revised** | **Description** | **Revised by Name and Title** | **Approved by Name and Title** |
| 0.1 | 03/06/2017 | Draft | Initial Draft | XXXX XXX | XXXX XXX |
| 0.2 | 3/28/2017 | Various sections | Different sections revised | XXX | XXX |
| 0.3 | 4/13/2017 | All | External Review | XXX | XXX |

**Table of Contents**

[**1**](#_heading=h.gjdgxs) **PURPOSE 3**

[**2**](#_heading=h.41mghml) **REFERENCE Documents 3**

[**3**](#_heading=h.tyjcwt) **Test Plan Overview 4**

[3.1 Project Description 4](#_heading=h.4d34og8)

[3.2 Test Objectives 4](#_heading=h.2s8eyo1)

[3.3 Assumptions 5](#_heading=h.17dp8vu)

[**4**](#_heading=h.3rdcrjn) **Risk AND MITIGATION 6**

[**5**](#_heading=h.26in1rg) **TEST STAGES 6**

[**6**](#_heading=h.lnxbz9) **Test Scope 8**

[**7**](#_heading=h.1ksv4uv) **Test Approach 8**

[7.1](#_heading=h.44sinio) PBI000000022893 - XXX Pool Number Numeric issue post XXX 14.1 release 8

[7.2](#_heading=h.2jxsxqh) PBI000000023896 - XXX Lender Loan # validation issue post XXX 14.1 release 9

[7.3](#_heading=h.2grqrue) Adjust reports and interfaces (new/change/retire). 9

[7.4](#_heading=h.vx1227) XX Download impacts 10

[7.5](#_heading=h.z337ya) XX UI Impacts 10

[7.6](#_heading=h.3fwokq0) Notification impacts 10

[7.7](#_heading=h.1v1yuxt) Change reporting cycle to daily submission of monthly reports to allow earlier reporting of loan-level data and corrections and provide more time to perform pool / security activities; aligns S/S and A/A loan cycles. 12

[7.8](#_heading=h.1y810tw) RTI- Real Time Interfaces Testing: 13

[**8**](#_heading=h.2xcytpi) **Test Data / Preparation 14**

[**9**](#_heading=h.3whwml4) **Test Environments 14**

[**10**](#_heading=h.2bn6wsx) **TEST TOOLS 14**

[**11**](#_heading=h.qsh70q) **DEFECT REPORTING AND TRACKING 14**

[**12**](#_heading=h.3as4poj) **ENTRY & EXIT CRITERIA 17**

[**13**](#_heading=h.2p2csry) **TRAINING (OPTIONAL) 17**

[**14**](#_heading=h.147n2zr) **COMMUNICATION AND STATUS REPORTING 17**

[**15**](#_heading=h.3o7alnk) **Test Schedule 17**

[**16**](#_heading=h.23ckvvd) **Defect Management Process 18**

[**17**](#_heading=h.32hioqz) **Roles & ReSPONSIBiLITIEs 18**

[**18**](#_heading=h.4f1mdlm) **Appendix** Error! Bookmark not defined.

[18.1 Glossary of Terms 19](#_heading=h.2u6wntf)

# PURPOSE

The Test Plan describes the objectives, scope, and approach of intended testing activities. It identifies what items (systems, subsystems, major functional areas, etc.) are being tested, what types of tests are being performed, as well as roles and responsibilities. The Test Plan is a comprehensive release-level document that covers all test activities planned (System, User, etc) in a single plan. The intent is to ensure management has visibility into the total scope of testing, responsibilities as well as any gaps or risks.

Testing activities must take place in a non-production environment. Testing may be executed by the IT team, business or both teams. For testing conducted by IT staff, the IT Owner must ensure the Segregation of Duties (SOD) standards are adhered to (for example, only authorized individuals shall have access to the testing system(s)/platform(s)). Project teams shall consult with the Project Sponsor and the CRB to determine the appropriate testing environment, strategy, tools, and methods to ensure a quality delivery.

The goal of testing is to ensure that changes to the production environment achieve specified business requirements and do not compromise the integrity of the data/system or change the processing functionality in a way that was unintended. At the project team’s discretion, any of the following combination of testing types shall constitute certification testing: System Testing, Regression Testing, User Acceptance Testing, Integration Testing, Volume Testing, Performance/Load Testing, Operational Testing, Information Security Testing, Conversion Testing and/or Compliance Testing. If upstream or downstream systems, applications or business processes are affected by the change, validate the change with these systems, applications and business processes and any associated interfaces. Additional tests may be conducted to validate documentation, training, contingency plans, disaster recovery, and installation depending upon the specific circumstances of the project.

This document meets the minimum set of required elements. Project teams may modify this template as necessary; however, it must contain all the required elements listed in this template. If an element is not mandatory for your change type, please do not delete that element. Simply note that element is “N/A” for your change type.

# Test Plan Overview

This document defines the specific test strategy, approach and processing necessary to perform validation activities for the XXX 15.1 Release. This test plan defines the scope and describes the preparation, execution and completion requirements to conduct User Acceptance Testing, including specifications for the appropriate database environment, execution sequence, defect management and staffing for the project.

## 

## 3.1 Project Description

XXX is a web-based application which provides XX-level information to Servicers and is available at www.efanniemae.com from 8:00 a.m. to 10:00 p.m. eastern standard time, Monday through Saturday. XXX provides online viewing of Servicer portfolio data for the current month plus three historical months of data as well. Some of the loan servicing functionality most commonly used includes View Loan, File Upload, XXX Files, SCRAMS Files, WLR Files, XXX Rejects, Notifications, Variable Rate Variable Payment, Maturity Rate Reduction, Downloads, Reports, Reconciliation and View Transmission functionalities around loan servicing.

XXX provides a reconciliation tool which will facilitate customers meet Fannie Mae’s loan reporting requirements by providing various data to do their reconciliation. Additionally, XXX allows users to correct their XXX a loan at a time or they can upload a file for bulk correction.

The objective of the Work stream is to deliver XXX functionality necessary to provide additional reporting capability on XXX UI and vend loan-level data reporting from XXX to Investor Reporting (XXX). Reconciliation data is required for servicer loan-level drafts. XXX will utilize this loan-level data to support its essential business functions.

Each month, servicers deliver loan payment transaction data and other loan data to XXX through XXX. That data must be provided to XXX reguXXXly in order for Master Servicing to perform its essential functions. The scope of the XX recently changed to reflect moving cutoff to an earlier date for loan-level data reporting from Servicers and additional reporting capability for Servicers. As a result, new XXX/XXX interfaces must be created to support the new reporting needs, XXX and XXX will support the negotiation and design of those new interfaces and provide the application functionality needed to deliver the defined data sets.

Changes to the Work stream’s scope must be made known to the Work stream’s resources as early as possible in the change management process so that the Work stream can adjust as needed and minimize the impact of any such change on any deliverables.

In order to support the Call-In Elimination program within Common Securitization Platform Integration (CSPi), the Master Servicing capability will adopt a standard Loan Reporting Cycle for all remittance types, accept daily liquidation reports, and determine draft amounts for loans based on applied loan activity.

## 3.2 Test Objectives

The Test objectives of ***XXX BAU*** **& *XX*** release are as follows:

1. To ensure that the XXX system meets business requirements as outlined in the in-scope Stakeholder Requirements and Project Scope.
2. To exercise all test cases and scenarios which support requirements as defined by the in-scope Stakeholder Requirements, mitigating the risk of application failure due to the changes introduced by this release.
3. To validate the in-scope ***XXX BAU & XX*** changes via validation of the User Interface behavior and targeted execution of XXX.
4. To validate that the end-users can continue to use the system to accomplish their defined objectives.
5. To execute a targeted set of regression tests to confirm the integrity of related in-scope functionality. These regression tests will focus on in-scope functionality and not on application’s performance.
6. To conduct Integration tests with XXX application, verifying that data can be passed from application to application and ensuring that processing of data/files upstream/downstream is completed per specification.
7. To confirm user functionality is not adversely impacted by the technical changes.

## 3.3 Assumptions

1. Prior to the start of UAT, the Development teams responsible for the design and coding of the system have performed Unit and Component Integration tests of the revisions.
2. The UAT Test Scope is limited to documented functionality outlined in the baseline version of the Stakeholder Requirements. Requirement updates, enhancements, and changes to existing documented/undocumented functionality and/or design have been outlined and reviewed. Any addition to the existing scope could impact UAT schedule.
3. UAT Team will have access to dedicated test environment(s) and testers will be granted access rights to the environments that match the security group(s) established for the End User roles.
4. Entrance criteria are met prior to the start of UAT execution.
5. Business SME support will be available to UAT Team during test planning and execution.
6. Technology Mandate changes to the Acceptance Environment will be deployed and verified by Tech Ops prior to start of UAT. Environment setup at the infrastructure level will be tested and supported by the respective development, database, and infrastructure teams.
7. The test data need to be as recent as possible.
8. UAT testing scope will exclude some technical and non-functional requirements. Items which cannot be verified by UAT testing process will be identified and approved for exclusion prior to publishing the Test Summary Document.

# Risk AND MITIGATION

The project level risks are maintained at the following EPPM location:

|  |  |  |
| --- | --- | --- |
| **Risk** | **Probability/ Impact** | **Prevention / Resolution** |
| Any environment downtime due to infrastructure changes may impact testing schedule. | Probability: Low  Impact: High | 1) UAT Team should validate the availability of the environments prior to the start of Testing.  2) If any environment is unavailable, UAT team will contact the Support and the Development teams for restoration as soon as possible. |

# TEST STAGES

**Requirements Testing:**

For Waterfall types of projects detailed documented requirements are necessary. It also reflects the needs of the business accurately prior to moving through the SDLC lifecycle. The primary objective of Requirements Testing is to ensure that Requirements are clear, complete, reasonably detailed, cohesive, attainable, and testable. Validation of requirements prior to the development and test stages with upfront quality control measures will reduce defects downstream and enhance cost savings. Requirements’ testing is a formal validation of the Test Input cases by the Test Teams to identify gaps, track and resolve defects and ultimately produce clear and accurate information downstream.

A two-step approach to requirements testing / review shall be used:

1. Stakeholders and key area leads shall review the Business Requirements Document – concentrating on: requirement completeness, requirement cXXXity and requirement testability. All comments and deXXXed modifications will be communicated to the Business Requirements team for inclusion in the BRD.
2. A walkthrough and review of the project Business Requirements Document shall be completed with all appropriate stakeholders, area leads and approvers. Again, participants will offer feedback and changes again confirming requirement – completeness, cXXXity and testability.

**Development Testing:**

Development will perform Unit Testing of the components being impacted in the functionality of the XXX application as part of the release. The development unit test plan will cover test scenarios for all requirements in the Business Requirements.  Unit test results will validate against the expected results for each of the requirements.

Development team will perform unit/string tests to confirm functionality, completeness and correctness of their coding efforts. Unit/string tests should be completed prior to offering code drop to the System Test environment.

**System Testing:**

System test will perform testing on functional changes to the application within the targeted scope of changes. System Test validation will follow black box testing methodology, under which the functionality of an application is validated as opposed to its internal structures or workings.

The goals of System Test are:

* + To plan and effectively execute System Test in order to validate solution specifications by exercising its components in an end-to-end customized execution sequence.
  + To provide an independent comprehensive functional assurance that prioritized system requirements are indeed met.
  + To provide assurance that the developed components are free of critical and high severity defects upon exit from System Test.
  + To partner with the Business Requirement team that verifies test artifact quality at all stages of test lifecycle from analysis to test execution.

**User Acceptance Testing (Business Acceptance Testing):**

The primary objective of User Acceptance Testing (UAT) is to ensure that the system functional changes have been implemented according to Stakeholder Requirements, without adversely impacting the existing system functionality. User Acceptance Testing validates that the end-users will be able to utilize the system to accomplish their defined objectives.

The UAT team will perform tests to confirm functional validity of the system/application within the targeted scope of changes. User Acceptance Testing should be completed prior to offering code drop to the Production environment and to provide assurance that the developed components are free of critical and high severity defects upon exit from User Acceptance Testing.

User Acceptance testing will be conducted in Approved Acceptance Environment.

**Integration Testing:**

Integration testing will verify that data can be passed from application to application to ensure that proper processing of data/files upstream or downstream (where applicable) are completed. Integration testing is determined per project based on the business impact of the change. For all S/IR interfaces impacted in a release, UAT will conduct integration testing with sending/receiving applications in a methodology to be mutually defined and approved with interface partners. Where feasible, this testing will include production-like data content and delivery simulation.

**Real Time Interface Testing:**

Per XXX mandate, all “Real Time Interfaces” must be tested for each SDLC application change. For cXXXification, a “Real Time Interface” is an interface that provides results in real-time to the requesting system.  Per this updated Interface Testing Requirement, when an application change is made, the project team must ensure coordination for testing with all upstream / downstream systems that are connected to the application using a Real Time Interface.

RTI testing with XXX:

* View Loans(Loan Finder)
* Loan Reporting(Loan Finder)
* Enhanced Downloads

**Performance Testing:**

For this release Performance testing is not in scope.

# Test Scope

**Progression Test Scope:**

**XX Scope Items*:***

1. Adjust reports and interfaces (new/change/retire).

See more details in section 7.3

**BAU Scope Items*:***

1. XXXX - XXX Pool Number Numeric issue post XXX 14.1 release
2. XXXX - XXX Lender Loan # validation issue post XXX 14.1 release
   1. **Requirements that cannot be certified**

UAT will perform regression testing for the following tech mandate items:

1. Upgrade Web logic Server from version 10.3.5 to 10.3.6 version.
2. Upgrade JDK version from version 1.6.0\_71 to version 1.7.0\_72

# Test Approach

The following is a high level approach of User Acceptance Testing (UAT) for the XXX 15.1 Release. User Acceptance Testing for XXX is focused on Functional (Progression) scenarios referenced by the approved project scope. The UAT team produces the Test Plan for review and approval, produces Test cases for review by stakeholders, and produces test data. Once code migration is complete, the UAT team performs a shakeout of the environment. The UAT team executes Targeted test cases followed by Regression testing. Additionally, several related functional areas have been included for Targeted Regression testing. During execution the UAT team performs Quality Control checks on the completed test cases and when the testing has successfully completed the team produces the required test results and summary documents.

**Progression Test Approach:**

### XXX - XXX Pool Number Numeric issue post XXX 14.1 release

1. Validate the XXX Reject Screen: Users will be able to input an alphanumeric value in the "Pool" field and "Servicer Loan Number" field. It will be tested with different data scenarios for this field.
2. Users will be able to input alphanumeric characters into the "Servicer Loan" field in the Notifications screen in XXX. It will be tested with different data scenarios for this field.

### XXX - XXX Lender Loan # validation issue post XXX 14.1 release

1. The system will populate the "Servicer Loan" field on the XXX Screen in XXX with data from the XXX application
2. The system will not validate the value populated in the "Servicer Loan" field on the Loan Reporting Screen in XXX.
3. UAT will test to validate the screen representation reflects the database value.

### XX UI Impacts

UAT team is going to validate the XX XXX UI changes for the following screens:

1. Header text is listed as ‘S/S, A/A and S/A Edit Update’ combined based on reporting cycle change, instead of ‘S/S Edit Update’ and ‘A/A & S/A Update’ listed individually.
2. In the Scheduling Dates section of the Admin screen, the Remittance types for Cut-Off Date and Update End Date are combined and shown as S/S, A/A, and S/A Update Cut-Off Date and S/S, A/A, and S/A Update End Date.
3. Text in footer of the screen has been updated by removing the word "LASER". The footer is

changed to “Data current as of: MM/DD/YYYY".

1. Payment/Rate Projection Notifications, Rate Reduction Changes, and Final Maturity Due Notifications dates will be updated as the system gets updated.
2. Header of the PDF generated file, accessible from the Formattable Version link on the View Loan screen, has the Cut off date combined to say 'S/S, A/A & S/A Cut-Off'.

## 

## Regression Testing

UAT will conduct a Risk Based Regression test of the business critical functionality. Risk-based Regression testing scope is defined based on inputs from the Business, Development, System Test Teams, Requirements, and documented via the Testing Risk Assessment workbook.

**The following UI components will be tested as part of Targeted Regression testing:**

1. Validate Portfolio summary
2. Validate MBS3+ Purchase Advice
3. Validate Modification Recap Report
4. Validate PFP Reclass
5. Validate PFP Book
6. Validate Maturity Report
7. Validate Rate Reductions
8. Validate Trial Balance Report
9. Validate Remittance Update Report
10. Validate Lender Recap Report
11. Validate Monthly Payment Note Rate Pass Thru Rate Change Form
12. Validate Final Maturity Due Report
13. Validate SHORTAGE/SURPLUS Analysis Report
14. Validate Action Code Reject Report
15. Negative Testing

* Invalid Input Validation in View Loan
* Invalid file upload
* Virus File Upload

The following batch jobs will be tested as part of regression:

1. AV#box#PFPNewIssue
2. AV#box#maturityDueLoad
3. AV#box#MaturityLoanRec
4. AV#box#rateReduction
5. AV#box#armLoans
6. AV#box#modificationRecapLoad
7. AV#box#portfolioSummaryLoad
8. AV#box#remittanceCorrection
9. AV#box#shortageSurplus
10. AV#box#fixLoans
11. AV#box#shortageSurplusAdjDtl
12. AV#box#PFPBook
13. AV#box#PFPReclass
14. AV#box#purchaseAdvice
15. AV#box#indexLoad
16. AV#box#PIMBSload
17. AV#box#rejectRecord
18. AV#box#RejectLoanRec
19. AV#box#VrvpLoanRec
20. AV#box#variableRate
21. AV#box#remittanceDetail
22. AV#box#XXXD300\_Processing
23. AV#box#MorningStatusPage
24. AV#box#checkForFilePresence
25. AV#box#deleteDownloadedORAbandonedFiles
26. AV#box#downloadFilesProcessor
27. AV#box#onlineXXX81-82-83-96Corrections
28. AV#box#rejCumExtract
29. AV#box#startXXXProcess

Attached below is the link for Risk Based Matrix. UAT developed this matrix gathering inputs from Business, SIT, Development, Requirements, and Production Support teams and have identified the high risk areas. Test cases will be prepared for the high risk areas if they do not exist already.

Risk Based Matrix Document

UAT will regression test the business critical regression test cases. UAT will validate that (1) business and technology identified critical batch jobs run successfully and completely, (2) the batch job logs are free of errors or warnings, and (3) the counts in the table match the counts with the loaded files.

**Integration Testing**:

Integration testing will verify that data can be passed from application to application to ensure that proper processing of data/files upstream or downstream (where applicable) are completed.  Integration testing is determined per project based on the business impact of the change. For all XXX interfaces impacted in a release, UAT will conduct integration testing with sending/receiving applications in a methodology to be mutually defined and approved with interface partners. Where feasible, this testing will include production-like data content and delivery simulation.

Integration testing will be performed with the following applications

|  |  |  |
| --- | --- | --- |
| **#** | **XXX Interface Designation** | **Source/Destination Partner Application** |
| 1 | WLR File Upload | Whole loan Remic (WLR) |
| 2 | EDI File Upload | MORNET |
| 3 | SCRAMS File Upload | SCRAMS |
| 4 | XXX File | XXX |

### RTI- Real Time Interfaces Testing:

UAT will perform Real Time Interface (RTI) testing with XXX. The following functionalities will be verified as part of RTI testing:

* Enhanced Downloads, INT967/G5610
* View Loans (use Loan Finder)
* Loan Reporting (use Loan Finder)

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Servicer Investor Reporting** | **Interfacing Application** | **Approach** |
| 1 | XXX | XXX | UAT will validate the field values from the XXX module and compare against the corresponding field values from XXX database. |

# Test Data / Preparation

UAT plans to have Test execution in acceptance environment. UAT will execute 1 full cycle of critical batch jobs as detailed in the Risk Based Matrix. UAT will take latest production backup and extract it to the AORZEN01 database for the UAT Acceptance Testing.

This approach will require the specific data setup:

1. Full Volume or lower environment
2. This environment will be used for XXX testing.
3. The test data in this environment will be initially prepared by restoring latest production backup.
4. Targeted batch execution will be performed for one full cycle of critical batch jobs.

# Test Environments

The test environments will include support for these features:

1. Full application services (On-Line U/I and Batch execution)
2. Full access privileges to the online environment and read-only access to the database schema
3. No new software tool requirements are anticipated or planned

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Environment Component | Component Location | Level of Security Needed | Special Testing Tools Needed | Additional Information |
| **User Acceptance Test** | **App Server Details:**  alasset-admin12a (admin),  alasset-ap12a, alasset-ap12b  **Batch Server Details:** acust-db01  **Database Server Details:** AOXXX01  **Operating System:** SoXXXis  **URL:** [XXX UI](https://faas.acptfanniemae.com/fed/idp/initiatesso?providerid=wl_SURF1035_acpt) | **Standard** | **JIRA** | **NA** |

# TEST TOOLS

The software tools listed below will be used:

|  |  |  |
| --- | --- | --- |
| **Software** | **Vendor** | **Use** |
| JIRA | Atlassian Corporation | Defect Tracking |
| JIRA | Atlassian Corporation | Scope, Requirements and Solution Specifications |
| Selenium | Selenium | Test Automation |
| MS Office | Microsoft | Test Plan, Test Rest Result Summary |

# DEFECT REPORTING AND TRACKING

Defect Management will be handled at the project level. Any and all defects will be recorded at the project level and documented in the testing system. Meetings will be held as necessary to determine any inter-project impact and the appropriate response/mitigation to the defect. All Program-related defects must be mitigated, resolved or accepted before production.

Defect tracking involves logging, monitoring, and tracking the status of system defects at the project level, and escalated as necessary to the program level. This includes any failure of the system to meet the business requirements or testing expectations.

Program Level defects will be reviewed by representatives of Development, System Test and UAT to determine the course of actions and appropriate resolution that is acceptable to the project and the program.

The following is the basic release and defect workflow:

1. Typically, the build schedule is defined by the development team and distributed to the Project team for review prior to System test execution. Builds are provided in accordance with the build schedule. Emergency builds proposed by Development for TEST are reviewed and coordinated by Configuration Management (CM) and System Test and UAT.
2. System Test/UAT/Business testers identify defects during testing. The defect is logged, with summary description, associated build version number, associated release id, primary functionality, severity, category etc., and assigned to a Reviewer (usually the Development Lead).
3. The reviewer assesses defects and assigns them to the appropriate developer on a reguXXX basis.
4. Developer designs enhancements or fixes, and then delivers code/stored procedures per build calendar. Project level developer completes and emails Release Notes to specific QA teams and changes defect status.
5. Final defect meeting is scheduled if necessary prior to PROD implementation to assess test coverage percentages, defect metrics and open issues for approval to migrate with the Project team.

Defects are initially rated for Severity Level by System Test/UAT and upon review, assigned a Priority status by the designated reviewer (i.e. Project Manager, Development Lead, and Business Lead).

**Defect Severities and Description**

The defect severity definition varies from one testing stage to another. The definitions are captured in the below table

|  |  |  |
| --- | --- | --- |
| **Severity**  **Description** | **Severity Definitions for System, User, and Production** | **Severity Definitions for Requirement Defects** |
| Critical | Severe business disruption, financial or reputational impact and no workaround exists. The customer is unable to use the product, resulting in a critical impact to their operation. This defect must be resolved before exiting current phase or releasing to production. | In-scope content is missing, or content is included that is not in scope, or content otherwise has major flaws that need to be corrected before it can be effectively reviewed. The Reason for the requirement defect is usually  Incomplete/Missing  Inconsistent  Incorrect |
| High | Significant business disruption but a workaround exists. The customer is able to use the product but is severely restricted. This defect should be resolved before exiting current phase or releasing to production. | Content has a major inaccuracy or is missing important detail. The Reason for the requirement defect is usually  Incomplete/Missing  Incorrect  Unclear  Inconsistent  Not traceable  Not testable |
| Medium | Minor business disruption of business, but has a workaround; minor usability issues. This defect should be resolved before exiting current phase or releasing to production. | Content is correct, but has a moderate flaw that needs amendment; for instance because it is unclear, imprecise, or not concise. Also used to note content that has missing or incorrect mapping to requirements. The Reason for the requirement defect is usually  Unclear  Not traceable  Not testable |
| Low | The defect may be cosmetic in nature; or a usability annoyance like warning messages; misspelled words, etc. | Formatting or organizational observation, or a grammatical or spelling error, not affecting meaning. The Reason for the requirement defect is usually  Unclear  Implementation dependant |

**Defect Priorities and Description:**

The defect priorities and definition vary from one testing stage to another. Their definitions are captured in the below table. The expected resolution timeframe for the defects depends on their priority

|  |  |  |
| --- | --- | --- |
| **Priority Description** | **Priority Definitions for defects in non-production** | **Priority Definitions for Production Defects** |
| Critical | Immediate attention – must receive highest development priority. This situation should be resolved immediately since the defect causes most if not all of the functional areas to be un-testable. | Immediate attention – must receive highest development priority. This should be resolved immediately |
| High | Should be reported immediately to the development team. A response or action plan must be provided within 2 working days since the defect causes more than one of the functional areas to be un-testable. | Should be reported immediately to the development team. A response or action plan must be provided within 2 working days. |
| Medium | A response or action plan should be reported within 5 working days. | A response or action plan should be reported within 5 working days. This defect should be resolved in the next release |
| Low | Fix dates are subject to negotiation. An action plan should be developed before the next release | Fix dates are subject to negotiation. An action plan should be developed for the next release |

# 

# ENTRY & EXIT CRITERIA

**Entrance:**

1. Requirements have been base lined and approved.
2. Successful unit testing is done and system testing must complete the targeted Testing.
3. Environment should be ready (e.g. shakeout of UAT environment performed successfully).
4. Test data should be available.
5. Any upstream/downstream dependenXXs have been satisfied.
6. Test Plan has been approved.
7. UAT Test Cases and Test Scripts have been approved.
8. Test Case walkthrough conducted to include testers, PM, DEV, CM, and other business stakeholders as required.
9. UAT Pre RTM is generated.
10. UAT Tester Analysts should have required access to the test environments.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Exit:**   1. There are no outstanding defects. 2. All test cases have been executed unless deferred. 3. The regression testing has been performed and completed. 4. Deferred test items have been identified, approved are moved to future release.(Defect severity: Low / Medium). 5. Retest and validation of non-deferred defects is complete. 6. Detailed Test Results have been reviewed and approved. 7. Final RTM and Test Case Specification documents are complete. 8. Stakeholder walkthroughs and approvals of Test Summary are complete.  TRAINING (OPTIONAL) N/A COMMUNICATION AND STATUS REPORTING Status reporting process followed at each test stage:   1. During UAT, daily status emails will be sent to all the stakeholders involved with the release. The status email will include a metrics of the Test Cases and Defects CQs. It will also include the progress of the test execution, validation and all the issues/CQs. 2. During the duration of the project DRB (Design Review Board) meetings will be held once a week to review all issues. Participants on this call include Stakeholders from Business, Technology, Requirement and Test Teams.  Test Schedule This test will be executed by the Delivery Support Services UAT Team in collaboration with the Single Family Technical Operations (Tech Ops) team. Additional support will be provided by Single Family DBA and Configuration Management (CM) teams.  The schedule below was developed to accommodate project time constraints at a high level. A detailed execution schedule will be used by the Project Management team to monitor progress against project objectives and the associated SDLC deliverables.   |  |  |  | | --- | --- | --- | | **Description** | **Start Date** | **End Date** | | Test Plan |  |  | | Develop Test Cases |  |  | | Progression – Test Case Execution |  |  | | ~~Progression – Test Case Execution (~~**~~Increment 3~~**~~)~~ |  |  | | Interface and RTI (Real Time Interface) Execution |  |  | | Regression Testing |  |  | | Complete Documentation |  |  | | Implementation |  |  |  Defect Management Process Standard issues tracking, reporting and resolution processes will be employed through the Clear Quest application and UAT Metrics reports.  Software defect reports are entered as Change Requests (CRs) and tracked during testing using the Rational Clear Quest defect-tracking tool. The CR Form is an automated process that is used to track all requests (including new features, enhancement requests, defects, etc.) along with related status information throughout the project lifecycle. A change history is maintained with each CR, including all state changes along with dates, names and reasons for the change. This information will be available for any repeat reviews and for final closing. On a release level, software defects are summarized and tracked for management purposes in a separate test report document.  Tests will either pass or fail. A test case fails if any one of the steps does not execute as stated. If step in a test case fails as test cases can contain several steps, due to software or documentation inconsistency, a CR is generated by the Tester to document the problem. The Tester is required to provide as much information as possible including any additional attachments that may help in problem duplication and resolution, such as the following:   1. Detailed description of the problem 2. Steps to reproduce 3. Expected results 4. Actual results 5. Reference documents 6. Environment |

# Roles & ReSPONSIBiLITIEs

|  |  |  |
| --- | --- | --- |
| **Role** | **Responsibility** | **Name of Team Member** |
| UAT Test Manager | Support UAT Lead and UAT analysts, as needed, to facilitate to efforts associated with planning, execution and validation of UAT testing. This includes reviewing of SDLC documentation before approvals. | XXX |
| UAT Test Lead | Overall responsibility for all Production/Ongoing UAT planning, execution, management, and status reporting. | XXX |
| Test Analyst(s) | Develop test cases, test scripts, and test data. Execute test cases (non-automated testing). Validate and document testing results. Compile and distribute defect reports. Coordinate resolution of defects. | XXX |
| UAT Test environment Setup | Responsible for Environment setup and batch execution on lower environments | XX XX |
| SIT Test Manager | Responsible for managing System test | XXX |
| SIT Test Lead | Responsible for SIT testing activities | XX XX |
| SIT Testers | Responsible for Planning, Execution and validation of the test scenarios and results. | XXX |
| Business Owner/ Manager | SME/Responsible for Business Requirements | XXX |
| SMEs/ Business Analysts | SME/Responsible for Business Analysis | XXXX |
| Technology Manager | Responsible for managing Development activities. | XXXX |
| Business Analysis Manager | Responsible for managing the systems analysis and requirements | XXX |
| Business Analyst | Responsible for creating Business Requirements Document | XXXX |
| Technology Lead | Responsible for code release and defect resolution. | XXXX |
| Technology Developers | Responsible for coding,  Release coding and analysis of defects. | XXX |
| Project Manager | Responsible for overall management of XXX 15.1 release | XXX |