Existing Product Intake Program (EPIP)

Patch PSS\*1.0\*203

Test Evaluation



Department of Veterans Affairs

**April 2017**

Version 2.0

Revision History

**Note**: The revision history cycle begins once changes or enhancements are requested after the Communications Plan has been baselined.

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| 04/04/2017 | 2.0 | Updated Test Results Summary and Defect Log due to retesting for re-remediation. | EPIP Project Team |
| 03/03/2017 | 1.0 | Initial document. | EPIP Project Team |

Artifact Rationale

The test evaluation document is the primary output of the test and evaluation process, an integral part of the systems engineering process, which identifies levels of performance and assists the developer in correcting deficiencies.

Table of Contents

[1. Test Evaluation Introduction 1](#_Toc478720727)

[1.1. Test Evaluation Scope 1](#_Toc478720728)

[1.2. Test Architecture 1](#_Toc478720729)

[1.3. Test Environment/Configuration 1](#_Toc478720730)

[1.4. Installation Process 2](#_Toc478720731)

[2. Test Data 2](#_Toc478720732)

[3. Issues 2](#_Toc478720733)

[4. Test Execution Log 2](#_Toc478720734)

[5. Test Defect Log 3](#_Toc478720735)

[6. Test Results Summary 3](#_Toc478720736)

[6.1. Defect Severity and Priority Levels 4](#_Toc478720737)

[6.2. Total Defects by Severity Level 4](#_Toc478720738)

[6.3. Breakdown of Test Results 4](#_Toc478720739)

[6.4. Performance Testing 4](#_Toc478720740)

[7. Test Coverage 4](#_Toc478720741)

[7.1. Requirements Covered 4](#_Toc478720742)

[7.2. Section 508 Compliance Coverage 5](#_Toc478720743)

[8. Suggested Actions 5](#_Toc478720744)

[9. Defect Severity and Priority Definitions 5](#_Toc478720745)

[9.1. Defect Severity Level 5](#_Toc478720746)

[9.1.1. Severity Level 1 – Critical 5](#_Toc478720747)

[9.1.2. Severity Level 2 - High 6](#_Toc478720748)

[9.1.3. Severity Level 3 - Medium 6](#_Toc478720749)

[9.1.4. Severity Level 4 - Low 6](#_Toc478720750)

[9.2. Priority Classifications 7](#_Toc478720751)

[9.2.1. Priority 1 - Resolve Immediately 7](#_Toc478720752)

[9.2.2. Priority 2 - Give High Attention 7](#_Toc478720753)

[9.2.3. Priority 3 - Normal Queue 7](#_Toc478720754)

[9.2.4. Priority 4 - Low Priority 7](#_Toc478720755)

[10. Optional Tables, Charts, and Graphs 7](#_Toc478720756)

[11. Document Approval Signatures 8](#_Toc478720757)

[Appendix A - Test Execution Log 9](#_Toc478720758)

[Appendix B – Defect Log 10](#_Toc478720759)

# Test Evaluation Introduction

The purpose of this Test Evaluation is to:

* Identify the testing approach used.
* Present a summary analysis of the key test results from the remediation of this intake for review and assessment by designated stakeholders.
* Provide a general statement of the quality of the system under test.
* Make recommendations for future testing efforts.

## Test Evaluation Scope

The scope of this Test Evaluation is to verify the functionality of the Patch PSS\*1.0\*203 code modification, as determined by Functional, Component Integration/System, and Regression testing. Testing activities followed the specifications outlined in the following Master Test Plan**: PSS\*1.0\*203** **Master Test Plan** (included in Appendix A).

## Test Architecture

Following are the EPIP test accounts used by the Leidos Development and SQA Testing teams to test PSS\*1.0\*203.

|  |  |
| --- | --- |
| **Development Test Accounts**  (For Unit Testing) | **SQA Test Accounts**  (For Functional, Regression, and Component Integration and System Testing) |
| VistAS1 ( alternate name: D1S1) | VistAG1 (alternate name: D1G1) |
| VistAS2 (alternate name: D1S2) – for CPRS GUI testing only | VistAG2 (alternate name: D1G2) – for CPRS GUI testing only |

## Test Environment/Configuration

The EPIP test accounts are maintained by the EPIP System Administrator, who installs all VA-released patches as soon as they are nationally released. All EPIP test accounts are cloned from existing VA Enterprise Testing Services (ETS) test accounts. The Computerized Patient Record System (CPRS) Graphical User Interface (GUI) executable is configured for each VistA instance utilizing a unique Internet Protocol (IP) address to connect to the VistA applications. Any updates to the CPRS GUI executable are handled by the EPIP System Administrator.

All EPIP Test Engineers and Developers who have the proper credentials can access the test accounts. The VA Austin Information Technology Center (AITC) support team resets passwords and sets up new access credentials on an as-needed basis.

## Installation Process

As soon as the remediation process is complete and the patch is available for testing, a KIDS build is created in the Development account and then sent to FORUM for final packaging. The patch is then submitted to the VA SQA Lead’s Mailman account for installation.

An EPIP Developer or Test Engineer utilizes the KIDS Installation process to extract the build from the patch and install the build into a test account. The individual who installs the patch verifies the routine checksums and also checks for errors during the installation process. If the patch is successfully installed without any errors, then the EPIP Test team proceeds with Functional, Regression, and Component Integration and System testing. If defects are found, then the Development team works to find a resolution and creates new versions of the patch until all defects are resolved.

# Test Data

The SQA Testing team utilizes the test data in the designated test accounts (D1G1, D1G2).

The test data is encrypted following the standards set forth by the VA Office of Information & Technology (OI&T). All Personally Identifiable Information (PII) and Protected Health Information (PHI) is scrubbed and is not available to the Test Engineers.

All testing is executed using encrypted test patients available from any of the EPIP test accounts. Examples of encrypted test patients:

AAAHURMMX, XPHY

BADHB, HAADXS

FDHUX, YHI J

All tests were executed manually by EPIP Test Engineers.

# Issues

No issues were encountered during testing of PSS\*1.0\*203.

| **Title** | **Issue Description** | **Type** | **Severity** |
| --- | --- | --- | --- |
| N/A | N/A | N/A | N/A |

# Test Execution Log

The Test Execution Log records the execution of test scripts and documents the test results for each test script.

The SQA Testing team utilizes the Rational Quality Manager (RQM) tool for all testing activities. All test documents are stored in the EPIP repository, including the Master Test Plan, Test Suites, Test Cases, and Test Scripts. Test execution is performed, and test results recorded, in RQM. The Test Engineer adds the test results to the Test Execution records to indicate whether testing achieved Pass or Fail status.

The Test Execution records for PSS\*1.0\*203 are included in the EPIP Patch PSS\*1.0\*203 Master Test Plan. The Master Test Plan is available in Appendix A.

# Test Defect Log

The Test Defect Log is a tool for recording, analyzing, tracking, and documenting the closure of defects. It specifies the screen, field, behavior or result that occurred, and the IEEE-defined Severity Level. It includes enough information for the developer to find and re-create the defect. The Defect Log is available in Appendix B.

# Test Results Summary

SQA testing for Test version 1 started in the Dev1 Gold1 test environment on December 27, 2016. Test version 1 was installed in this test environment after Unit testing in Dev1 Silver1 was completed. Upon completion of Integration testing (Component Integration and System Testing, Functional Testing, and Regression Testing), one (1) defect was found and reported. This defect was due to a missing build discovered during Functional testing.

A new version (v2) was then submitted to SQA for retesting. Test version 2 was installed in the Dev1 Gold1 test environment after Unit testing in Dev1 Silver1 was completed. Upon completion of Integration testing (Component Integration and System Testing, Functional Testing, and Regression Testing), two (2) defects were found and reported.

A new version (v3) was then submitted to SQA for retesting. Test version 3 was installed in the Dev1 Gold1 test environment after Unit testing in Dev1 Silver1 was completed. Testing continued until February 16, 2017. Upon completion of Integration testing (Component Integration and System Testing, Functional Testing, and Regression Testing), zero (0) defects were found and reported. During the Product demonstration, it was noted that a requirement was missing from the patch. One (1) defect was entered to address this issue.

Test version 4 was submitted and installed in the Dev1 Gold1 test environment on March 20, 2017 after Unit testing in Dev1 Silver1 was completed. Upon completion of Integration testing (Component Integration and System Testing, Functional Testing, and Regression Testing), one (1) defect was found, resulting in a new test version 5.

Test version 5 was installed on the same day as Test version 4 after Unit testing in Dev1 Silver1 was completed. Upon completion of Integration testing (Component Integration and System Testing, Functional Testing, and Regression Testing), one (1) defect was found.

Test version 6 was submitted to SQA and installed on March 21, 2017 after Unit testing in Dev1 Silver1 was completed. Upon completion of Integration testing (Component Integration and System Testing, Functional Testing, and Regression Testing), one (1) defect was found, which resulted in test version 7.

Test version 7 was installed in the Dev1 Gold1 test environment on March 22, 2017 after Unit testing in Dev1 Silver1 was completed. Upon completion of Integration testing (Component Integration and System Testing, Functional Testing, and Regression Testing) on March 30, 2017, zero (0) defects were found and reported.

## Defect Severity and Priority Levels

A defect is defined as a flaw in a component or system that can cause the component or system to fail to perform its required function, e.g., an incorrect statement or data definition. A defect, if encountered during execution, may cause a failure of the component or system.

Defects are categorized according to severity and priority levels. The test analyst assigns the severity, while the development manager assigns the priority for repair. For more information, see Defect Severity and Priority Definition in this Test Evaluation.

## Total Defects by Severity Level

The Defect Log in Appendix B displays the defects encountered while testing this patch, and the severity level of each.

## Breakdown of Test Results

Testing was completed on March 30, 2017*.* All test results were recorded in RQM. Detailed results are available in the EPIP Patch PSS\*1.0\*203 Master Test Plan (see Appendix A).

## Performance Testing

Performance testing was not conducted.

# Test Coverage

The EPIP Patch PSS\*1.0\*203 Master Test Plan contains details on test coverage (see   
Appendix A).

## Requirements Covered

The requirements for PSS\*1.0\*203 are stored in the Rational Dynamic Object Oriented Requirements System (DOORS). The test cases used to validate that the requirements have been addressed are stored in RQM and are linked to the requirements, providing full traceability. The user stories stored in the Rational Change and Configuration Management (CCM) application (a.k.a. Rational Team Concert (RTC)) are linked to the requirements and test cases.

The following links provide access to the various EPIP repositories in the Rational toolkit.

* EPIP (RM) – Go to **Artifacts**, then **Browse Artifacts**, and then search for the desired patch number. The patch folder contains the requirements for that patch.
* , and then search for the desired Master Test Plan. The Master Test Plan and test cases are linked to requirements.
* EPIP (CM) – Go to **Plans**, then **All Plans**, and then search for the desired sprint Plan. The user stories in each Plan are linked to requirements and test cases.

## Section 508 Compliance Coverage

Section 508 testing was not required for this patch.

# Suggested Actions

Leidos recommends moving this patch to IOC testing.

# Defect Severity and Priority Definitions

The classification of defects within a system examines both the severity and priority of the defect.

Severity is a measure of how great the impact is on the user’s ability to complete the documented actions within the system.

Priority determines the speed with which a given defect must be repaired.

Defect classification may be determined either because testing is delayed by a failure in the system or because a cumbersome workaround prevents a user from completing the assigned tasks. Both severity and priority measures must be recorded when scheduling defect resolution tasks.

## Defect Severity Level

The following subsections identify the defect severity levels.

### Severity Level 1 – Critical

Institute of Electrical and Electronics Engineers (IEEE) definition: The defect results in the failure of the complete software system, of a subsystem, or of a software unit (program or module) within the system.

* Any defect that compromises patient safety or system security. Examples of system security defects include breach of confidentiality requirements of the Privacy Act, the Health Insurance Portability and Accountability Act (HIPAA), or Federal Tax Information guidelines.
* Loss of system functionality critical to user operations with no suitable workaround, i.e., there is no way to achieve the expected results using the application.
* System crash or hang that prevents further testing or operation of the complete application or a section of the application.
* Any defect that causes corruption of data from a result of the system (as opposed to user error).
* Any defect in which inappropriate transmissions are consistently generated or appropriate transmissions of HL7 messages fail to be generated.
* Loss of functionality resulting in erroneous eligibility/enrollment determinations or communications not being sent.

### Severity Level 2 - High

IEEE definition: The defect results in the failure of the complete software system, of a subsystem, or of a software unit (program or module) within the system. There is no way to make the failed component(s) function. However, there are acceptable processing alternatives which will yield the desired result.

* A major defect in the functionality that does not result in corruption of data.
* A major defect in the functionality resulting in a failure of all or part of the application, where:
* The expected results can temporarily be achieved by alternate means. The customer indicates the work around is acceptable for the short term.
* Any defect that does not conform to Section 508 standards.
* Any defect that results in inaccurate or missing requirements.
* Any defect that results in invalid authentication or authentication of an invalid end user.

### Severity Level 3 - Medium

IEEE definition: The defect does not result in a failure, but causes the system to produce incorrect, incomplete, or inconsistent results, or the defect impairs the systems usability.

* Minor functionality is not working as intended and a workaround exists but is not suitable for long term use
* The inability of a valid user to access the system consistent with granted privileges
* Typographical or grammatical errors in the application, including installation guides, user guides, training manuals, and design documents
* Any defect producing cryptic, incorrect, or inappropriate error messages
* Any defect that results from the use of non-standard data terminology in the application or documentation, as defined by the Department of Veterans Affairs
* Cosmetic issues that are important to the integrity of the product, but do not result in data entry and or data quality problems.

### Severity Level 4 - Low

IEEE definition: The defect does not cause a failure, does not impair usability, and the desired processing results are easily obtained by working around the defect.

* Minor loss of, or defect in the functionality where a long term use exists
* Low-level cosmetic issues.

## Priority Classifications

The following subsections identify the appropriate actions for defects at each priority level, per definitions of IEEE.

### Priority 1 - Resolve Immediately

Further development and/or testing cannot occur until the defect has been repaired. The system cannot be used until the repair has been affected.

### Priority 2 - Give High Attention

The defect must be resolved as soon as possible because it is impairing development and/or testing activities. System use will be severely affected until the defect is fixed.

### Priority 3 - Normal Queue

The defect should be resolved in the normal course of development activities. It can wait until a new build or version is created.

### Priority 4 - Low Priority

The defect is an irritant that should be repaired, but can be repaired after more serious defects have been fixed.

# Optional Tables, Charts, and Graphs

None.

# Document Approval Signatures

Signed: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Program/Project Manager Date

Signed: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Business Sponsor Representative Date

Signed: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Test Lead Date

Appendix A - Test Execution Log

The Test Execution Records for PSS\*1.0\*203 are included in the EPIP Patch PSS\*1.0\*203 Master Test Plan.

Appendix B – Defect Log

Seven (7) defects were found during testing of versions 1 through 7. All defects have been resolved.

| **Defect ID** | **Affected Screen** | **Affected Field** | **Observed Behavior** | **Severity** | **Description** |
| --- | --- | --- | --- | --- | --- |
| N/A | None | None | N/A | High | No defects were found during Unit Testing of version 1.0. |
| N/A | N/A | N/A | N/A | N/A | No defects were found during Component Integration/System Testing of version 1.0. |
| N/A | N/A | N/A | N/A | N/A | No defects were found during Regression Testing of version 1.0. |
| 467060  Missing required build | N/A | N/A | N/A | N/A | One (1) defect was found during Functional Testing of version 1.0. |
| N/A | N/A | N/A | N/A | N/A | No defects were found during Unit Testing of version 2.0. |
| N/A | N/A | N/A | N/A | N/A | No defects were found during Component Integration and System Testing of version 2.0. |
| N/A | N/A | N/A | N/A | N/A | No defects were found during Regression Testing of version 2.0. |
| 457539 | Pharmacy Mail message | Mailman | No message received for Audit | High | Two (2) defects were found during Functional Testing of version 2.0. |
| 458406 | Pharmacy Mail message | Mailman | No message received for Drug edit. | High |
| N/A | N/A | N/A | N/A | N/A | No defects were found during Unit Testing of version 3.0. |
| N/A | N/A | N/A | N/A | N/A | No defects were found during Component Integration and Unit Testing of version 3.0. |
| N/A | N/A | N/A | N/A | N/A | No defects were found during Regression Testing of version 3.0. |
| 492935 | Pharmacy Mail Message | Mailman | Missing the Drug Price functionality | Critical | One (1) defect was found during Functional testing of version 3.0.  Note: Although this defect was found during the product demonstration, it affected patch functionality and therefore is shown as a Functional test defect. |
| N/A | N/A | N/A | N/A | N/A | No defects were found during Unit Testing of version 4.0. |
| N/A | N/A | N/A | N/A | N/A | No defects were found during Component Integration and System Testing of version 4.0. |
| N/A | N/A | N/A | N/A | N/A | No defects were found during Regression Testing of version 4.0. |
| 492937 | Pharmacy Mail Message | Mailman | Dispense / Order Units field incorrect display | Critical | One (1) defect was found during Functional testing of version 4.0. |
| N/A | N/A | N/A | N/A | N/A | No defects were found during Unit Testing of version 5.0. |
| N/A | N/A | N/A | N/A | N/A | No defects were found during Component Integration and System Testing of version 5.0. |
| N/A | N/A | N/A | N/A | N/A | No defects were found during Regression Testing of version 5.0. |
| 492970 | Pharmacy Application | VistA programmer prompt | User cannot enter Price Per Order Unit | Critical | One (1) defect was found during Functional testing of version 5.0. |
| N/A | N/A | N/A | N/A | N/A | No defects were found during Unit Testing of version 6.0. |
| N/A | N/A | N/A | N/A | N/A | No defects were found during Component Integration and System Testing of version 6.0. |
| N/A | N/A | N/A | N/A | N/A | No defects were found during Regression Testing of version 6.0. |
| 492975 | Pharmacy Application | Pharmacy Application | Missing code/ functionality from the build | Critical | One (1) defect was found during Functional testing of version 6.0. |
| N/A | N/A | N/A | N/A | N/A | No defects were found during Unit Testing of version 7.0. |
| N/A | N/A | N/A | N/A | N/A | No defects were found during Component Integration and System Testing of version 7.0. |
| N/A | N/A | N/A | N/A | N/A | No defects were found during Regression Testing of version 7.0. |
| N/A | N/A | N/A | N/A | N/A | No defects were found during Functional Testing of version 7.0. |