**State Prescription Monitoring Program Enhancement**

Version 1.2

Master Test Plan



January 2016

Department of Veterans Affairs

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Author |
| 1/28/16 | 1.2 | Updated with test site information |  |
| 12/16/15 | 1.1 | Multi-Divisional section updated per feedback |  |
| 12/8/15 | 1.0 | Document Baseline |  |

Table of Contents

1. Introduction 1

1.1. Purpose 1

1.2. Test Objectives 1

1.3. Roles and Responsibilities 2

1.4. Processes and References 2

2. Items To Be Tested 3

2.1. Overview of Test Inclusions 3

2.2. Overview of Test Exclusions 3

3. Test Approach 3

3.1. Product Component Test 4

3.2. Component Integration Test 5

3.3. System Tests 5

3.4. User Functionality Test 6

3.5. Enterprise System Engineering Testing 7

3.6. Initial Operating Capability Evaluation 7

4. Testing Techniques 8

4.1. Risk-based Testing 8

4.2. Enterprise Testing 9

4.2.1. Security Testing 9

4.2.2. Privacy Testing 9

4.2.3. Section 508 Compliance Testing 9

4.2.4. Multi-Divisional Testing 9

4.3. Test Types 9

4.4. Productivity and Support Tools 11

5. Test Criteria 11

5.1. Process Reviews 11

5.2. Pass/Fail Criteria 11

5.3. Suspension and Resumption Criteria 12

6. Test Deliverables 12

7. Test Schedule 13

8. Test Environments 13

8.1. Test Environment Configurations 13

8.2. Base System Hardware 14

8.3. Base Software Elements in the Test Environments 14

9. Staffing and Training Needs 15

10. Risks and Constraints 15

11. Test Metrics 15

Attachment A – Approval Signatures 16

Appendix A - Test Type Definitions 17

# Introduction

The State Prescription Monitoring Program (SPMP) provides the Department of Veterans Affairs (VA) the ability to transmit controlled substances prescription drug data to state-level reporting databases known as Prescription Drug Monitoring Programs (PDMPs) or Prescription Monitoring Programs (PMPs). The intended users of the enhancements or Patient Safety Issues (PSI) fixes are pharmacists, pharmacy technicians, and Veteran patients and their families. The goal of this enhancement supports improved business process, patient care, and resource utilization, as well as reduces spending and/or increases revenue generation.

## Purpose

The purpose of the Master Test Plan for the SPMP Enhancement project is to:

* Provide a central artifact to govern the planning and control of the test effort for the SPMP Enhancement project. It defines the general testing approach the team will follow. This is the top-level plan that will be used by managers to govern and direct the detailed testing work.
* Provide visibility into the testing efforts for stakeholders.

This document will also support the following objectives:

* Identify the functional requirements and user stories utilized for testing.
* Identify the assumptions, risks and constraints that affect this testing process.
* Outline the testing schedule.
* Describe the testing strategy, activities, and tools.
* Identify the Test Sites.
* Include the roles and responsibilities of the resources participating in the testing.
* Document the pass/fail performance of the software design parameters.

## Test Objectives

This Master Test Plan supports the following objectives:

To provide test coverage for 100% of the documented requirements within the approved Requirements Specification Document (RSD) and user stories.

To execute 100% of the test cases during System Testing and User Functional Testing.

To create, maintain, and control the test environment.

Establishes that the requirements have been implemented and are functioning correctly.

Outlines the testing approach that will be used, including the progression of testing coverage and references to quality milestones/gates.

Identifies the required resources.

Lists the deliverable elements of the test project.

Identifies defect reporting processes.

Describes the system to be tested, the test inclusions and exclusions, the roles, and resource participation for test and configuration management.

## Roles and Responsibilities

Table 1 lists the key roles and their responsibilities for this Master Test Plan.

Table 1: Roles and Descriptions

| Role | Description |
| --- | --- |
| Project Manager | Person that has overall responsibility for the successful planning and execution of a project; person responsible for creating the Master Test Plan in collaboration with the SQA Test Analyst(s). |
| Development Team | Persons that build or construct the product/product component. |
| Stakeholders | Persons that hold a stake in a situation in which they may affect or be affected by the outcome. |
| SQA Test Lead | An experienced SQA Test Analyst that leads and coordinates activities related to all aspects of testing based on an approved Master Test Plan and schedule. |
| SQA Test Analyst | Person responsible for ensuring full execution of the test process to include the verification of technical requirements and the validation of business requirements. |
| Test Team | A team made up of SQA Test Analysts and a SQA Test Lead that are responsible for executing the test plan and reporting test progress. |

## Processes and References

The processes that guide the implementation of this Master Test Plan are:

* Test Preparation
* Product Build
* Independent Test and Evaluation

The references that support the implementation of this Master Test Plan are:

* ProPath
* VA Section 508
* [Privacy Impact Assessment - Privacy Service](http://www.privacy.domain/Privacy_Impact_Assessment.asp)
* [Requirements Specification Documents (RSD) Version 1.2, November 2015](http://domain.ext.domain.ext/warboard/anotebk.asp?proj=1849&Type=Active%20)
* [System Design Document (SDD) Version 1.0, October 2015](http://domain.ext.domain.ext/warboard/anotebk.asp?proj=1849&Type=Active%20)

# Items To Be Tested

All functional requirements documented within Section 3.6 of the SPMP RSD will be tested within the Master Test Plan.

Please refer to the approved Requirement Specification Document (RSD) located in the [Technical Services Project Repository (TSPR) project notebook for SPMP](http://domain.ext.domain.ext/warboard/anotebk.asp?proj=1849&Type=Active%20).

## Overview of Test Inclusions

The Test Analyst will develop test scripts related to each specific requirement and user story related to the functionality of the SPMP Enhancement project.

## Overview of Test Exclusions

The following components, features, and combinations of components and features will not be tested:

* Requirements that fall outside the SPMP Enhancements project scope.
* Testing of functionality within the VistA Pharmacy package that was not impacted by the SPMP Enhancement project.
* Data flow testing of existing VistA applications.
* Development within the VistA application done by other project teams

# Test Approach

The SPMP Enhancement project will utilize a modified Agile Methodology. The epics, user stories, and requirements developed in each sprint will be documented in the RSD and in IBM Rational Team Concert (RTC). The SQA Analysts will work closely with the Business Analysts and Developers to develop test cases and test scripts as per the sprint schedule. The test cases and test scripts will be documented in the test deliverables in the form of word documents. The test cases will be mapped to the requirements in the Requirements Traceability Matrix (RTM). Test results will be captured within test scripts and reported via the RTM and Test Evaluation Summary. The project’s SharePoint and RTC will be utilized to track any issues/defects discovered during testing. Through this modified agile process, evolving prototypes of components will eventually stabilize in design and attain a release-worthy level to be included for national release.

The following test environments will be used:

* **INP namespace in DDEVVISTA** – VMS environment that will be used by the SPMP development team for development of the code and for Product Component Testing.
* **TSP namespace in DDEVVISTA** – VMS environment that will be used by the SPMP SQA team for conducting Component Integration Testing (CIT) and System Testing.
* **VAAUSSPMAPPDEV1** – Linux environment that will be used by the SPMP SQA team for conducting CIT and System Testing.
* **Oklahoma PMP Test User Account** – Test account owned by the Oklahoma PMP that will be used by the VA to test transmitting and validate the result of test transmissions for Product Component Testing, CIT, System Test as well as User Functional Test.
* **SQD** – VMS environment that will be used by the SPMP SQA Analyst to perform the SQA Checklist Reviews.

Testing will include the following:

* ***Systematic Testing*** – Execution of test cases designed for test coverage with expected results.
* ***Negative Testing*** – Execution of test cases designed to produce unexpected results for test coverage where practical.
* ***Ad-hoc/Exploratory Testing*** – A type of testing that is informal and improvisational to assess the viability of a features or products.
* ***Regression Testing*** – A type of testing that validates existing functionality still performs as expected when new functionality is introduced into the system under test.
* ***Feature Testing*** – A type of testing performed by the SQA Analyst for each sprint to verify that the feature was completed.

## Product Component Test

Prior to delivery of the sprint build to the SQA Test Team, the SPMP developers will perform Product Component Testing, also known as Unit Testing, on code they have developed. This testing includes the internal technical and functional testing of a module/component of code, and verifies that the requirements defined in the SDD have been successfully applied to the module/component under test. This testing will take place in the Albany development environment. Upon successful result, the developer will inform the SQA analyst that the new feature is ready to be tested and will provide instructions for testing. Steps may include:

* Analyze requirements to understand the application functionality and dependencies
* Identify all the routines affected by the module or object
* Specify all the routines that are called from various locations
* Execute tests on prioritized options
* Execute tests with different combinations of options and data. For example, test with minimal data entered and test with maximal data entered
* Perform exploratory testing, i.e., randomly exercise the module, object, and options based on domain knowledge, past performance, and expertise
* Record the actual test
* Perform M Code Secondary Developer's Review Checklist

## Component Integration Test

After the development team has completed Product Component Testing for a sprint, the development team will deliver the build to the SQA Test Team to perform Component Integration Testing. This will take the form of manual tests to perform smoke testing to confirm the stability and viability of the delivered build to proceed to system testing. The SQA Team and the Project Manager will make an assessment based on those test results whether to continue with system testing or return to development for additional work.

The SQA Test Analyst installs the Product Component, performs the Component Integration Test (CIT) within the TSP test environment as follows:

1. The SQA Test Analyst will perform CIT to expose defects in the interfaces and interaction between integrated components. CIT will verify the patch installation and the testing environment. The installation will be installed into the testing environment ensuring the installation process does complete and does not produce any errors.
2. CIT will consist of a subset of the test scripts and will cover critical path application functionality to ensure the application is functioning properly after the installation of the patch software. This testing also verifies that the application is stable and not prone to errors due to the new software installation.
3. The SQA Test Analyst will record the test result. If there is a defect found, the SQA Test Analyst will send an email to the Developers with the test results and continue performing the smoke test, if possible. The severity of the defect will be assessed by the team as to whether testing can proceed to System Testing. If it is deemed necessary to resolve the defect prior to proceeding to System Test, then a new patch will be created with the fix and it will go back through product component test and CIT.

## System Tests

Agile methodology is being followed for development, so the development team delivers a new build to the SQA Test Team frequently, as new builds become available and stable enough to promote to test. System testing will be executed by the SQA Test Team. Test scripts will be created for each sprint based on identified user stories and requirements being delivered for the sprint. Testing will take the form of exercising the test scripts when appropriate and additional “ad hoc” testing when necessary. Transmissions can be simulated using the Oklahoma box and the resulting transmitted reports can be retrieved and evaluated. The actual transmission process to a SPMP is not possible to validate outside of a production account. Test results will be delivered to the Project Manager after each sprint has completed being tested. These results will be in the form of a Test Execution Log, Defect Log and a Test Evaluation Summary.

The SQA Test Analyst performs the System Test within the TSP test environment as follows:

1. The SQA Test Analyst will perform System Test in TSP Test Environment.

2. The SQA Test Analyst will record test results in within the test script documentation.

When System Test is complete:

1. Defects will be logged in RTC following the Work Instructions provided by the CM team.
2. Testing Result reports are generated and distributed (frequency to be determined).
3. The SQA Test Analyst will perform the SQA Review Checklist.
4. End of testing reports are generated and distributed along with the completed SQA Review Checklist, and Test Evaluation Summary.

System testing ends unless defects are reported during UFT or IOC and a new product build needs to be tested. In this case, the test cycle begins again starting with product component test, followed by component integration test, and system test.

## User Functionality Test

User Functional Testing (UFT) will be performed upon completion of System Testing. UFT Testing will be performed for the sprint by stakeholders within the test sites’ test account, testing the functionality of the application. Transmissions can be simulated using the Oklahoma box and the resulting transmitted reports can be retrieved and evaluated. The actual transmission process to a SPMP is not possible to validate outside of a production account. Test scripts will be provided to the test sites from the SQA Test Team, and the stakeholders will provide results for reporting progress and completion of UFT by delivering completed test scripts back to the SQA Test Team. A collaboration software portal, or SharePoint site, will be utilized that the test sites and the SPMP development team can access. This will allow for easy communications with all parties involved with UFT as well as defect reporting. The SQA Test Analyst will document UFT test results and deliver the documentation to the Project Manager, in the form of a UFT Test Defect Log, UFT Test Execution Log and UFT Test Evaluation Summary as per the documented process in ProPath. The content of these documents will be derived from the test results the stakeholders provide from their completed test scripts.

Table 2: Test Sites and Points of Contact

| Test Site | Contacts |
| --- | --- |
| Bedford, MA-VAMC | , Pharmacy ADPAC  , Information Technology Specialist |
| Boston Regional HCS | , Pharmacy ADPAC  , Information Technology Specialist |
| Cheyenne, WY-VAMC | , Pharmacy ADPAC |
| Fort Harrison, MT | , Pharmacy ADPAC  / , Local OI&T Contacts |
| Hines, IL-VAMC | , Pharmacy ADPAC |
| Indianapolis, IN | , Pharmacy ADPAC  , Local OI&T Contact |
| Iron Mountain, MI-VAMC | , Pharmacy ADPAC |
| Louisville, KY-VAMC | , Pharmacy ADPAC |
| Northampton, MA (Western Central MA HCS) | , Pharmacy ADPAC  / , Local OI&T Contacts  , Information Technology Specialist |
| San Diego, CA-VAMC | , Pharmacy ADPAC |
| Syracuse, NY (VISN 2) | (SYR) , Pharmacy ADPAC  (VISN) , Pharmacy ADPAC  , Information Technology Specialist |
| Tennessee Valley HCS | , Pharmacy ADPAC  , Pharmacy ADPAC  , Local OI&T Contact |
| Big Springs, TX (West Texas HCS) | , Pharmacy ADPAC  , Local OI&T ADPAC |

## Enterprise System Engineering Testing

Enterprise System Engineering (ESE) testing will be conducted by the ESE Testing Services team and supported by the SPMP Enhancement development team. The SPMP Enhancement Project Manager will conduct a kick off meeting with ESE Testing Services to plan and coordinate any needed support for the operational readiness review performed by the ESE Testing Services team.

## Initial Operating Capability Evaluation

Initial Operating Capability (IOC) testing will be performed for the increment upon completion of UFT after the SPMP Enhancement Project Manager has submitted the IOC entry request form and approval received. Testing will be performed at the test sites in their individual production accounts, using the same test scripts provided during UFT. Production testing will continue for a minimum of two weeks. The stakeholders will provide results for reporting progress and completion of IOC by delivering completed test scripts back to the SQA Test Team. A collaboration software portal, or SharePoint site, will be utilized that the test sites and the SPMP development team can access. This will allow for easy communications with all parties involved with IOC as well as defect reporting. Each test site will respond via MS Outlook and provide written approval of the increment when IOC testing has been completed for that site. The SQA Test Analyst will document IOC test results and deliver the documentation to the Project Manager after each increment has been tested, in the form of an IOC Test Defect Log, IOC Test Execution Log, and IOC Test Evaluation Summary as per the documented process in ProPath. The content of these documents will be derived from the test results the stakeholders provide via the collaboration software portal.

# Testing Techniques

Testing techniques include both static and dynamic testing. Static analysis focuses on appropriate methods that are used to determine or estimate software quality without reference to actual executions.

Static testing techniques include the following:

* Review of business requirements (RSD)
* Review of functional specifications and design documents
* Review of user stories
* Preparation of test plan
* Preparation of test scenarios and test cases
* Execution of walkthroughs and inspections

Dynamic analysis deals with specific methods for ascertaining software quality through actual executions (i.e. with real data and under real circumstances). Dynamic testing techniques include:

* Product Component Testing
* System Testing
* Regression Testing
* Product Integration Testing
* Usability Testing
* End to End Testing
* User Functionality Testing

## Risk-based Testing

The SPMP SQA Analyst will test identified risks throughout each testing cycle.

## Enterprise Testing

This project does not address any enterprise level requirements, therefore this testing is not required.

### Security Testing

This project does not address any specific security requirements, therefore this testing is not required.

### Privacy Testing

This project does not address any specific privacy requirements, therefore this testing is not required.

### Section 508 Compliance Testing

The SPMP development team is responsible for ensuring that the SPMP functionality is usable from the keyboard and is compliant with Section 508 requirements. The SPMP development team will perform 508 testing using Microsoft Eyes provided by the 508 office and submit the appropriate self-certification forms when necessary to ensure the SPMP application is 508 compliant.

The SPMP team will use the following checklists as a guide when building the application and provide these as completed checklists as the result of internal 508 testing, along with the self-certification and conformance validation forms. We will work closely with the 508 office to ensure that the SPMP functionality within the legacy VistA application adheres to the current guidelines.

* 1194.21 Software applications and operating systems Word document
* 1194.31 Functional performance criteria Word document
* 1194.41 Information, documentation, and support Word document

### Multi-Divisional Testing

This project does contain Multi-Divisional Testing.  There are sites that have multiple divisions spread across more than one state.  Tests will be run during CI/ST where multiple divisions will be set up for multiple states and transmissions validated.

## Test Types

Table 3: Test Types

| Test Types | Party Responsible |
| --- | --- |
| Access control testing | n/a |
| Benchmark testing | n/a |
| Build verification testing | Development Team |
| Business cycle testing | n/a |
| Compliance testing | ESE, 508 Office, Development Testing, Test Team |
| Component integration testing | Development Team |
| Configuration testing | n/a |
| Contention testing | n/a |
| Data and database integrity testing | Test Team |
| Documentation testing | Test Team |
| Error analysis testing | Test Team |
| Exploratory testing | Development Team, Test Team |
| Failover testing | n/a |
| Installation testing | Development Team |
| Integration testing | Development, Test Team |
| Load testing | n/a |
| Migration testing | n/a |
| Multi-divisional testing | n/a |
| Parallel testing | n/a |
| Performance monitoring testing | n/a |
| Performance testing | n/a |
| Privacy testing | n/a |
| Product component testing | Developer |
| Recovery testing | n/a |
| Regression test | Test Team |
| Risk based testing | Test Team |
| Section 508 compliance testing | Development Testing, Test Team, 508 Office |
| Security testing | n/a |
| Smoke testing | Development Testing |
| Stress testing | n/a |
| System testing | Test Team |
| Usability testing | n/a |
| User Functionality Testing | Stakeholders |
| User interface testing | Test Team, Test Sites |

## Productivity and Support Tools

Table 3 describes the tools that will be employed to support this Master Test Plan.

Table 4: Tool Category or Types

| Tool Category or Type | Tool Brand Name | Vendor or In-house | Version |
| --- | --- | --- | --- |
| Defect Tracking | Rational Team Concert (RTC) | IBM | 4.0 |
| Project Management | Rational Change and Configuration Management (CCM) | IBM | 4.0.5 |
| Configuration Management | Rational Change and Configuration Management (CCM) | IBM | 4.0.5 |
| Requirements Management | Rational Requirements Composer | IBM | 4.0.5 |
| Epic and User Story Management | Rational Team Concert (RTC) | IBM | 4.0 |

# Test Criteria

## Process Reviews

The Master Test Plan under goes two reviews:

* Peer Review – upon completion of the Master Test Plan
* Formal Review – after the Project Manager approves the Master Test Plan

For more information on the reviews associated with testing, see the Product Build, Test Preparation, and Independent Test and Evaluation processes.

## Pass/Fail Criteria

Incidents identified during the execution of this test plan will be evaluated to determine their severity. This impact will be recorded in the severity section of the defect report.

* (1) High Impact Test Incident is an error or lack of functionality that:
  + Jeopardizes patient or personnel safety by corrupt or incorrect data
  + Has no workaround to provide similar functionality and this functionality is required to move to system, integration, or user acceptance
  + Adversely affects all users or key user functionality
* (2) Medium Impact Test Incident is an error or lack of functionality that:
  + Has a reasonable workaround to maintain functionality
  + Impacts a small group of users, but has workaround
  + Functionality works but not to requirements, specifications, or standards and workflow is not hampered
* (3) Low Impact Test Incident is an error or lack of functionality that may cause operator/user inconvenience and minimally affects operational processing.
  + Spelling errors
  + Minor GUI Graphical/Formatting errors that do not affect functionality/visibility
* (4) Enhancement Test Incident is something that would be “nice” to have in the integration piece but was not included in the specifications for this release.

All High and Medium defects shall be addressed or negotiated prior to release. Any limitation or outstanding test incident shall have an approved contingency process (workaround) in place.

## Suspension and Resumption Criteria

Testing will cease on a test item when a high impact test incident is logged. Testing will resume when the incident is addressed.

Testing will cease on the entire release when three high impact test incidents are logged. Testing will resume when the incidences are addressed.

Testing will cease if any element of the test system is unavailable, such as the VistA system.

# Test Deliverables

Table 4 lists the test deliverables for the SPMP project.

Table 5: Test Deliverables

| Test Deliverables | Responsible Party |
| --- | --- |
| Master Test Plan | , SQA Analyst |
| Test Cases/Test Scripts | and , SQA Analysts |
| Test Environment | and , SQA Analysts |
| Test Evaluation Summaries | and , SQA Analysts |
| Traceability Report or Matrix | and , SQA Analysts |

# Test Schedule

The overall project schedule is being managed within the Project Plan.

Refer to the [TSPR project notebook for SPMP](http://domain.ext.domain.ext/warboard/anotebk.asp?proj=1849&Type=Active%20) for the current project plan.

The following test phases will be completed prior to national release.

* Integration/System Testing
* User Functionality Testing
* Operational Readiness Testing
* Initial Operating Capability Testing

Table 6: Testing Milestones

| Testing Milestones | Responsible Party |
| --- | --- |
| Complete Master Test Plan | , SQA Analyst |
| Complete test cases and/or scripts | and , SQA Analysts |
| Complete Requirements Traceability Matrix | and , SQA Analysts |
| Complete SQA testing | and , SQA Analysts |

# Test Environments

Successful testing requires control of the test environment. Unplanned changes to the test environment may introduce new test incidents, alter the expected test results, and thus invalidate the test cases. Successful testing requires controlled access to the test environment, an environment that replicates the field environment as closely as possible.

## Test Environment Configurations

Development testing will be conducted in a dedicated VistA development environment. Testing will be conducted in a dedicated VistA testing environment. All promotions to the test environment will be managed through a change management process and access to this account will be managed by the Test Team.

The VistA development environment will on the Albany Office of Information Field Office (OIFO) system and the VistA test environment will reside on the Bay Pines OIFO system. The SPMP development and SQA test teams are responsible for configuring and maintaining these environments, including installing newly released VistA patches. Developers and SQA Test Analysts will be reviewing and confirming test data.

## Base System Hardware

Table 6 sets forth the system resources for the test effort presented in this Master Test Plan.

The specific elements of the test system may not be fully understood in early iterations, so this section may be completed over time. The test system should simulate the production environment as closely as possible, scaling down the concurrent access and database size, and so forth, if and where appropriate. Tailor the System Hardware Resources table as required.

Table 7: System Hardware Resources

| Resource | Quantity | Name and Type |
| --- | --- | --- |
| NT Workstation |  | Windows XP or greater |
| VistA database type |  | Cache |

## Base Software Elements in the Test Environments

Table 7 describes the base software elements that are required in the test environment for this Master Test Plan.

Table 8: Software Elements

| Software Element Name | Version |
| --- | --- |
| Inpatient Medications | 5.0 |
| Kernel | 8.0 |
| Mailman | 8.0 |
| National Drug File | 4.0 |
| Outpatient Pharmacy | 7.0 |
| Pharmacy Data Management | 1.0 |
| RPC Broker (32-bit) | 1.1 |
| Toolkit | 7.3 |
| VA FileMan | 22.0 |

# Staffing and Training Needs

There are no specific training needs at this time. This section will be updated as training needs arise.

Refer to the [SPMP Project Management Plan](http://domain.ext.domain.ext/warboard/anotebk.asp?proj=1849&Type=Active%20) for additional information on staffing and training.

# Risks and Constraints

The risk log is captured in the [SPMP Project Management Plan](http://domain.ext.domain.ext/warboard/anotebk.asp?proj=1849&Type=Active%20). For additional information on risks, please see the Risk Management standards in ProPath.

# Test Metrics

Metrics are a system of parameters or methods for quantitative and periodic assessment of a process that is to be measured.

Test metrics may include, but are not limited to:

* Number of test cases (pass/fail)
* Percentage of test cases executed
* Number of requirements and percentage tested
* Percentage of test cases resulting in defect detection
* Number of defects attributed to test case/test script creation
* Percentage of defects identified; listed by cause and severity
* Time to re-test

The Final Test Evaluation Summary Report completed by the SQA Test Analyst captures the measures specified above.

# Attachment A – Approval Signatures

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

Date

Program Manager / IPT Chair

Product Development

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

, Pharm D Date

Business Sponsor

Associate Chief Consultant for

Clinical Informatics and Pharmacy Re-engineering for

, RPh, MHSA, Chief Consultant

Pharmacy Benefits Management

VHA Office of Patient Care Services

# Appendix A - Test Type Definitions

| Test Type | Definition |
| --- | --- |
| Access Control Testing | A type of testing that attests that the target-of-test data (or systems) are accessible only to those actors for which they are intended, as defined by use cases. Access Control Testing verifies that access to the system is controlled and that unwanted or unauthorized access is prohibited. This test is implemented and executed on various targets-of-test. |
| Benchmark Testing: | A type of performance testing that compares the performance of new or unknown functionality to a known reference standard (e.g., existing software or measurements). For example, benchmark testing may compare the performance of current systems with the performance of the Linux/Oracle system. |
| Build Verification Testing  (Prerequisite: Smoke Test) | A type of testing performed for each new build, comparing the baseline with the actual object properties in the current build. The output from this test indicates what object properties have changed or don’t meet the requirements. Together with the Smoke test, the Build Verification test may be utilized by projects to determine if additional functional testing is appropriate for a given build or if a build is ready for production. |
| Business Cycle Testing | A type of testing that focuses upon activities and transactions performed end to end over time. This test type executes the functionality associated with a period of time (e.g., one-week, month, or year). These tests include all daily, weekly, and monthly cycles, and events that are date-sensitive (e.g., end of the month management reports, monthly reports, quarterly reports, and year-end reports). |
| Capacity Testing | [Capacity](http://www.geekinterview.com/question_details/48768) testing occurs when you simulate the number of users in order to stress an application's hardware and/or network infrastructure. Capacity testing is done to determine the capacity (CPU, Data Storage, LAN, WAN, etc.) of the system and/or network under test. |
| Compliance Testing | A type of testing that verifies that a collection of software and hardware fulfills given specifications. For example, these tests will minimally include: “core specifications for rehosting – ver.1.5-draft 3.doc”, Section 508 of The Rehabilitation Act Amendments of 1998, Race and Ethnicity Test, and VA Directive 6102 Compliance. It does not exclude any other tests that may also come up. |
| Component Integration Testing | Testing performed to expose defects in the interfaces and interaction between integrated components as well as verifying installation instructions. |
| Configuration Testing | A type of testing concerned with checking the programs compatibility with as many possible configurations of hardware and system software. In most production environments, the particular hardware specifications for the client workstations, network connections, and database servers vary. Client workstations may have different software loaded, for example, applications, drivers, and so on hand, at any one time; many different combinations may be active using different resources. The goal of the configuration test is finding a hardware combination that should be, but is not, compatible with the program. |
| Contention Testing | A type of performance testing that executes tests that cause the application to fail with regard to actual or simulated concurrency. Contention testing identifies failures associated with locking, deadlock, livelock, starvation, race conditions, priority inversion, data loss, loss of memory, and lack of thread safety in shared software components or data. |
| Data and Database Integrity Testing | A type of testing that verifies that data is being stored by the system in a manner where the data is not compromised by the initial storage, updating, restoration, or retrieval processing. This type of testing is intended to uncover design flaws that may result in data corruption, unauthorized data access, lack of data integrity across multiple tables, and lack of adequate transaction performance. The databases, data files, and the database or data file processes should be tested as a subsystem within the application. |
| Documentation Testing | Documentation testing is a type of testing that should validate the information contained within the software documentation set for the following qualities: compliance to accepted standards and conventions, accuracy, completeness, and usability. The documentation testing should verify that all of the required information is provided in order for the appropriate user to be able to properly install, implement, operate, and maintain the software application. The current VistA documentation set can consist of any of the following manual types:  Release Notes, Installation Guide, User Manuals, Technical Manual, and Security Guide. |
| Error Analysis Testing | This type of testing verifies that the application checks for input, detects invalid data, and prevents invalid data from being entered into the application. This type of testing also includes the verification of error logs and error messages that are displayed to the user. |
| Exploratory Testing | A technique for testing computer software that requires minimal planning and tolerates limited documentation for the target-of-test in advance of test execution, relying on the skill and knowledge of the tester and feedback from test results to guide the ongoing test effort. Exploratory testing is often conducted in short sessions in which feedback gained from one session is used to dynamically plan subsequent sessions. |
| Failover Testing | A type of testing test that ensures an alternate or backup system properly “takes over” (i.e., a backup system functions when the primary system fails). Failover Testing also tests that a system continually runs when the failover occurs, and that the failover happens without any loss of data or transactions. Failover Testing should be combined with Recovery Testing. |
| Installation Testing | A type of testing that verifies that the application or system installs as intended on different hardware and software configurations, and under different conditions (e.g., a new installation, an upgrade, and a complete or custom installation). Installation testing may also measure the ease with which an application or system can be successfully installed, typically measured in terms of the average amount of person-hours required for a trained operator or hardware engineer to perform the installation. Part of this installation test is to perform an uninstall. As a result of this uninstall, the system, application and database should return to the state prior to the install. |
| Integration Testing | An incremental series of tests of combinations or sub-assemblies of selected components in an overall system. Integration testing is incremental in a successively larger and more complex combinations of components tested in sequence, proceeding from the unit level (0% integration) to eventually the full system test (100% integration). |
| Load Testing | A performance test that subjects the system to varying workloads in order to measure and evaluate the performance behaviors and abilities of the system to continue to function properly under these different workloads. Load testing determines and ensures that the system functions properly beyond the expected maximum workload. Additionally, load testing evaluates the performance characteristics (e.g., response times, transaction rates, and other time-sensitive issues). |
| Migration Testing | A type of testing that follows standard VistA and HeV-VistA operating procedures and loads the latest .jar version onto a live copy of VistA and HeV-VistA. The following are examples of the types of tests that can be performed as part of migration testing:   * Data conversion has been completed * Data tables are successfully created * Parallel test for confirmation of data integrity * Review output report, before and after migration, to confirm data integrity * Run equivalent process, before and after migration |
| Multi-Divisional Testing | A type of testing that ensures that all applications will operate in a multi-division or multi-site environment recognizing that an enterprise perspective while fully supporting local health care delivery. |
| Parallel Testing | The same internal processes are run on the existing system and the new system. The existing system is considered the “gold standard”, unless proven otherwise. The feedback (expected results, defined time limits, data extracts, etc.) from processes from the new system are compared to the existing system. Parallel testing is performed before the new system is put into a production environment. |
| Performance Monitoring Testing | Performance profiling assesses how a system is spending its time and consuming resources. This type of performance testing optimizes the performance of a system by measuring how much time and resources the system is spending in each function. These tests identify performance limitations in the code and specify which sections of the code would benefit most from optimization work. The goal of performance profiling is to optimize the feature and application performance. |
| Performance Testing | Performance Testing assesses how a system is spending its time and consuming resources. Performance testing optimizes a system by measuring how much time and resources the system is spending in each function. These tests identify performance limitations in the code and specify which sections of the code would benefit most from optimization work. Performance testing may be further refined by the use of specific types of performance tests, such as, benchmark test, load test, stress test, performance monitoring test, and contention test. |
| Performance – Benchmark Testing | A type of performance testing that compares the performance of new or unknown functionality to a known reference standard (e.g., existing software or measurements). For example, benchmark testing may compare the performance of current systems with the performance of the Linux/Oracle system. |
| Performance – Contention Testing | A type of performance testing that executes tests that cause the application to fail with regard to actual or simulated concurrency. Contention testing identifies failures associated with locking, deadlock, livelock, starvation, race conditions, priority inversion, data loss, loss of memory, and lack of thread safety in shared software components or data. |
| Performance – Endurance Testing | Endurance testing, also known as soak testing, is usually done to determine if the system can sustain the continuous expected load. During soak tests, memory utilization is monitored to detect potential leaks. |
| Performance – Load Testing | A performance test that subjects the system to varying workloads in order to measure and evaluate the performance behaviors and abilities of the system to continue to function properly under these different workloads. Load testing determines and ensures that the system functions properly beyond the expected maximum workload. Additionally, load testing evaluates the performance characteristics (e.g., response times, transaction rates, and other time-sensitive issues). |
| Performance - ProfilingTesting | Performance profiling assesses how a system is spending its time and consuming resources. This type of performance testing optimizes the performance of a system by measuring how much time and resources the system is spending in each function. These tests identify performance limitations in the code and specify which sections of the code would benefit most from optimization work. The goal of performance profiling is to optimize the feature and application performance. |
| Performance – Spike Testing | A performance test in which an application is tested with sudden increment and decrements in the load. The focus is on system behavior during dramatic changes in load. |
| Privacy Testing | A type of testing that ensures that (1) veteran and employee data are adequately protected and (2) systems and applications comply with the Privacy and Security Rule provisions of the Health Insurance Portability and Accountability Act (HIPAA). |
| Product Component Testing | Product Component Testing (aka Unit Testing) is the internal technical and functional testing of a module/component of code. Product Component Testing verifies that the requirements defined in the detail design specification have been successfully applied to the module/component under test. |
| Recovery Testing | A type of testing that causes an application or system to fail in a controlled environment. Recovery processes are invoked while an application or system is monitored. Recovery testing verifies that application or system, and data recovery is achieved. Recovery Testing should be combined with Failover Testing. |
| Regression Test | A type of testing that validates existing functionality still performs as expected when new functionality is introduced into the system under test. |
| Risk Based Testing | A type of testing based on a defined list of project risks. It is designed to explore and/or uncover potential system failures by using the list of risks to select and prioritize testing. |
| Section 508 Compliance Testing | A type of test that (1) ensures that persons with disabilities have access to and are able to interact with graphical user interfaces and (2) verifies that the application or system meets the specified Section 508 Compliance standards. |
| Security Testing | A type of test that validates the security requirements and to ensure readiness for the independent testing performed by the Security Assessment Team as used by the Assessment and Authorization Process. |
| Smoke Test | A type of testing that ensures that an application or system is stable enough to enter testing in the currently active test phase. It is usually a subset of the overall set of tests, preferably automated, that touches parts of the system in at least a cursory way. |
| Stress Testing | A performance test implemented and executed to understand how a system fails due to conditions at the boundary, or outside of, the expected tolerances. This failure typically involves low resources or competition for resources. Low resource conditions reveal how the target-of-test fails that is not apparent under normal conditions. Other defects might result from competition for shared resources (e.g., database locks or network bandwidth), although some of these tests are usually addressed under functional and load testing. Stress Testing verifies the acceptability of the systems performance behavior when abnormal or extreme conditions are encountered (e.g., diminished resources or extremely high number of users). |
| System Testing | System testing is the testing of all parts of an integrated system, including interfaces to external systems. Both functional and structural types of testing are performed to verify that the system performance, operation and functionality are sound. End to end testing with all interfacing systems is the ultimate version. |
| Usability Testing | Usability testing identifies problems in the ease-of-use and ease-of-learning of a product. Usability tests may focus upon, and are not limited to: human factors, aesthetics, consistency in the user interface, online and context-sensitive help, wizards and agents, user documentation. |
| User Functionality Test | User Functionality Test (UFT) is a type of Acceptance Test that involves end-users testing the functionality of the application using test data in a controlled test environment. |
| User Interface Testing | User-interface (UI) testing exercises the user interfaces to ensure that the interfaces follow accepted standards and meet requirements. User-interface testing is often referred to as GUI testing. UI testing provides tools and services for driving the user interface of an application from a test. |

Template Revision History

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| June 2015 | 1.16 | Updated metadata to show record retention information and required by PMAS, VHA Release Management, Enterprise Operations, and VistA Intake Program | Process Management |
| May 2015 | 1.15 | Reordered cover sheet to enhance SharePoint search results | Process Management |
| March 2015 | 1.14 | Miscellaneous updates including the addition of Performance testing. | Channing Jonker |
| November 2014 | 1.13 | Updated to latest Section 508 conformance guidelines and remediated with Common Look Office Tool | Process Management |
| August 2014 | 1.12 | Removed requirements for ESE Approval Signature | Process Management |
| October 2013 | 1.11 | Converted to Microsoft Office 2007-2010 format | Process Management |
| July 09, 2012 | 1.10 | Added System Design Document to Section 1.2 -Test Objectives as an example | Process Management |
| January 03, 2012 | 1.9 | Updated Approval Signatures for Master Test Plan in Appendix a | Process Management |
| October 13, 2011 | 1.8 | Replaced references to Test and Certification with Independent Test and Evaluation. Replaced references to Certification and Accreditation with Assessment and Authorization. | Process Management |
| October 4, 2011 | 1.7 | Repaired link to Privacy Impact Assessment | Process Management |
| August 23, 2011 | 1.6 | Changed Operational Readiness Testing (ORT) to Operational Readiness Review (ORR) | Process Management |
| April 12, 2011 | 1.5 | Updated the Signatory Authorities in Appendix A in light of organizational changes | Process Management |
| February 2011 | 1.4 | Removed Testing Service Testing and Operational Readiness Testing; added Enterprise System Engineering Testing.  Changed Initial Operating Capability Testing to Initial Operating Capability Evaluation | Process Management |
| January 2011 | 1.3 | Repaired broken link in section 1.4 | Process Management Service |
| August 2010 | 1.2 | Removed OED from template | Process Management Service |
| December 2009 | 1.1 | Removed “This Page Intentionally Left Blank” pages. | OED Process Management Service |
| July 2009 | 1.0 | Initial ProPath release | OED Process Management Service |