**Alert Watch and Response Engine (AWARE)**

Master Test Plan



May 2014

Prepared by Harris Corporation

Revision History

| Creation Date | Version No. | Description/Comments | Author(s) | Reviewer(s) | Review Type | Issue Date |
| --- | --- | --- | --- | --- | --- | --- |
| 01/03/2014 | 0.01 | Initial | S. Greenacre | B. Jan B. Tjahjo L. Glance | Peer | 05/27/2014 |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

Table of Contents

[1 Introduction 1](#_Toc388965211)

[1.1 Purpose and Scope of Document 1](#_Toc388965212)

[1.2 Test Objectives 1](#_Toc388965213)

[1.3 Roles and Responsibilities 1](#_Toc388965214)

[1.4 Project Processes and References 2](#_Toc388965215)

[2 Items to be Tested 3](#_Toc388965216)

[2.1 Overview of Test Inclusions 3](#_Toc388965217)

[2.2 Overview of Test Exclusions 3](#_Toc388965218)

[3 Test Approach 4](#_Toc388965219)

[3.1 Test Phases 4](#_Toc388965220)

[3.2 Entry and Exit Criteria 4](#_Toc388965221)

[3.3 Test Data 5](#_Toc388965222)

[3.4 Test Suite/Cases/Script Specifications 5](#_Toc388965223)

[3.5 Defect Reporting and Tracking 5](#_Toc388965224)

[3.5.1 Defect Specification 5](#_Toc388965225)

[3.5.2 Severity Definitions 5](#_Toc388965226)

[3.5.3 Defect Resolution Process 6](#_Toc388965227)

[3.6 Metrics and Reporting 7](#_Toc388965228)

[3.7 Testing Tools 7](#_Toc388965229)

[3.8 Testing Risks and Mitigation 8](#_Toc388965230)

[3.9 Product Component Testing 8](#_Toc388965231)

[3.10 Component Integration Testing 8](#_Toc388965232)

[3.11 System Tests 9](#_Toc388965233)

[3.12 User Functionality Test (UFT) 9](#_Toc388965234)

[3.13 Testing Service Test 9](#_Toc388965235)

[3.14 Operational Readiness Testing Support 10](#_Toc388965236)

[3.15 Initial Operating Capability Testing 10](#_Toc388965237)

[4 Testing Techniques 10](#_Toc388965238)

[4.1 Risk-Based Testing 10](#_Toc388965239)

[4.2 Enterprise Testing 10](#_Toc388965240)

[4.2.1 Security Testing 10](#_Toc388965241)

[4.2.2 Privacy Testing 10](#_Toc388965242)

[4.2.3 Section 508 Compliance Testing 10](#_Toc388965243)

[4.2.4 Multi-Divisional Testing 10](#_Toc388965244)

[4.3 Test Types 11](#_Toc388965245)

[4.4 Iterative Testing 11](#_Toc388965246)

[4.5 Productivity and Support Tools 11](#_Toc388965247)

[5 Test Criteria 11](#_Toc388965248)

[5.1 Process Reviews 11](#_Toc388965249)

[5.2 Pass/Fail Criteria 11](#_Toc388965250)

[5.3 Suspension and Resumption Criteria 12](#_Toc388965251)

[5.4 Acceptance Criteria 12](#_Toc388965252)

[6 Test Deliverables 13](#_Toc388965253)

[7 Test Schedule 13](#_Toc388965254)

[8 Test Environments 13](#_Toc388965255)

[8.1 Test Environment Configurations 13](#_Toc388965256)

[8.2 Base System Hardware 13](#_Toc388965257)

[8.3 Base Software Elements in the Test Environments 14](#_Toc388965258)

[9 Staffing and Training Needs 14](#_Toc388965259)

[10 Constraints 14](#_Toc388965260)

[11 Acronyms 15](#_Toc388965261)

[12 Test Types Descriptions (and Minimum Set of Test Types Required) 15](#_Toc388965262)

[13 Attachment A - Approval Signatures 20](#_Toc388965263)

List of Tables

[Table 1 – Roles and Responsibilities 1](#_Toc388965264)

[Table 2 – Entry and Exit Criteria 4](#_Toc388965265)

[Table 3 – Tool Category or Types 8](#_Toc388965266)

[Table 4 – Test Types to be Performed 11](#_Toc388965267)

[Table 5 – Test Deliverables 13](#_Toc388965268)

[Table 6 – System Hardware Resources 13](#_Toc388965269)

[Table 7 – Software Elements 14](#_Toc388965270)

[Table 8 – Staffing Resources 14](#_Toc388965271)

[Table 9 – Acronyms and Abbreviations 15](#_Toc388965272)

[Table 10 - Test Types 16](#_Toc388965273)

# Introduction

## Purpose and Scope of Document

The purpose of the Master Test Plan (MTP) is to describe the overall testing approach in support of the Alert Watch and Response Engine (AWARE) project for the Department of Veterans Affairs (VA), VA Center for Innovation (VACI), and the Veterans Health Administration (VHA) Innovation Program. The MTP specifies the objectives, scope, test activities, test strategy, and assigned responsibilities for the test program of AWARE. The test strategy defines the test levels and types of tests planned throughout the second increment’s project life cycle.

The MTP provides a central artifact that governs the planning and execution of test endeavors. It defines the test approaches that will be employed to test the software and to evaluate the results of that testing. The MTP represents a top-level and detailed plan for use by management to govern and direct the overall test program for AWARE.

The MTP is a living document and will be updated as required.

## Test Objectives

The high-level test objectives for AWARE include:

* Provide test coverage for all testable functional requirements
* Execute 100 percent of the test cases during all applicable smoke, regression, and system testing
* Create, maintain and control the test environment
* Verify the software interfaces in the integrated systems
* Assist and support VA Beta and Pilot Site Testing
* Verify that the software interfaces correctly with existing systems
* Adapt to development methodology, testing for proper functionality as early as possible

## Roles and Responsibilities

Below, the primary roles and responsibilities as they pertain to this MTP, and the individual assigned to that role for AWARE are listed in Table 1.

Table 1 – Roles and Responsibilities

| Role | Responsibilities | Assigned To |
| --- | --- | --- |
| Harris Project Manager (PjM) | * Project planning * Management of project resources * Tracking and reporting of project status * Management of project risks and issues | Blake Jan |
| Harris Configuration Manager (CM) | * Provide and maintain a repository of change requests, code change requests, issues, risks, software release packages, and documentation. | Linda Glance |
| Development Team | * Provides the PjM with design, code, Product Component Testing, and deployment estimates * Provides assistance to the PjM and team in identification of risks and issues * Participates in technical reviews * Serves as a key point of contact for technical questions * Conducts the final review of technical documents * Performs coding of software changes for AWARE * Develops impact/risk assessments for upgrades * Generates the software build and verifies its components are included * Performs testing on changes to software units (Product Component Testing) * Responsible for planning and leading software deployments to test and production environments * Definition and implementation of structured and repeatable deployment processes * Provides assistance to testers and test sites to externally test software changes and updates * Provides fixes for assigned test incidents * Reports defect resolutions to the Test Engineer(s) * Provides peer reviews of documentation as needed * Provide team development support of product release plans | Harris / Astronaut |
| Test Team | * Provides PjM and planner(s) with test planning and conduct estimates * Completes testing of the features defined in the MTP * Creates test cases and scripts based upon documented requirements * Enters identified defects or problems into defect-tracking tool(s) * Performs follow-up to reported defects and problems found during test activities * Provides PjM and Tech Lead with summary report of test event results * Serves as point of contact to the AWARE teams for testing activities of the products * Captures risks associated with testing and tracks in the AWARE-assigned repository * Completes and distributes final test incidents reports * Participates in code reviews as necessary | Harris / Astronaut / User Works |

## Project Processes and References

The processes that guide the implementation of this MTP are:

* Harris Validation Process
* Harris Verification Process
* Harris Measurement and Analysis Process
* VA ProPath Test Preparation Process
* VA ProPath Product Build Process
* VA ProPath Test and Certification Process

The references that support the implementation of this MTP are:

* AWARE Requirements Specification Document (RSD)
* AWARE CPRS Integration Specification
* AWARE User Interface Document
* AWARE Requirement Traceability Matrix
* AWARE T4\_Test\_Cases
* VA Section 508 Office Web Page <http://www.section508.gov/>
* Office of Information & Technology (OIT) ProPath website <http://vaww.oed.oit.va.gov/process/propath/>
* VA Privacy Impact Assessment – Privacy Service <http://vaww.privacy.va.gov/PIA.asp>

# Items to be Tested

AWARE application test coverage will be provided for the functional requirements specified in the RSD.

## Overview of Test Inclusions

The Test Team will develop test cases and scripts specifically targeting each functional area. These test cases will be traced to specific requirements being implemented in the AWARE application under test. The AWARE RSD defines each requirement for which tests will be developed by the Test Team.

The following components and features will be tested:

* AWARE access and navigation
* AWARE administration
* AWARE GUI Knowledge-Base Editor
* AWARE SQL Tracking Database
* AWARE Business Rules Processing
* AWARE/CPRS Integration
* AWARE Quality Improvement (QI) Tool Functionality
* 508 compliance
* CPRS v29 only

## Overview of Test Exclusions

The following components and features will not be tested:

* VistA Applications that do not touch the VistA AWARE Package
* CPRS Versions below V29 or above V29
* Requirements outside AWARE
* Use cases outside AWARE
* Human Factors/Usability testing

# Test Approach

The overall testing approach is to follow the series of test events specified by ProPath. AWARE is an existing prototype software application developed by VACI. The existing AWARE prototype software and the application Enhancements are developed using a combination Agile Scrum and traditional Waterfall software development life cycle methodology.

Beta Readiness Testing will occur on the VACI Innovations Sandbox integrating with version 29 of CPRS during AWARE/CPRS Integration and prior to the formal testing required by the Project Management Accountability System (PMAS). This test verifies that the product meets all the requirements specified in the RSD for and does not collect any personal information from Veterans or their families. This testing is performed on the VACI Innovations Sandbox environment and is documented as stated in the AWARE PWS. Developers perform unit testing as each piece is completed. The Test Team first performs the system test informally to identify any defects. The Development Team resolves the defects as they are identified.

Following a system test event, AWARE goes through a series of test events specified by ProPath and run by VA. These include basic functionality, system integration, and 508 Compliance testing. The Harris team provides support and test scripts from System Testing for use during these events. The Harris team re-tests, verifies, and fixes any defects uncovered during this testing.

## Test Phases

The test phases follow the ProPath process as stated in the PWS. Additional details on these phases are available in sections 3.9 through 3.15.

## Entry and Exit Criteria

Entry and exit criteria are listed in below.

Table 2 – Entry and Exit Criteria

| Test Phase | Entry Criteria | Exit Criteria/Threshold |
| --- | --- | --- |
| Product Component Test | * Completion of code as it is developed | * Proof of exercise of all written code |
| Component Integration Test | * Completion of Product Component test for sprint * Successful completion of a smoke test | * Successful execution of integrated components |
| System Test | * Completion of all development before system test * Successful build into system test environment | * Successful execution of reviewed test cases/scripts, as documented in:   + RTM |

## Test Data

During the test preparation phase, the test team will request a copy of the Gold database from the VA CM team. The Gold database has been generated by the VistA Imaging team and consists of the following items:

* All of the nationally released Kernel Installation & Distribution System (KIDS) builds
* Test user accounts
* Test patients

The Developers and testers will create any additional test data that is required to test the functionality of AWARE.

## Test Suite/Cases/Script Specifications

The Test Team will create test cases/scripts using the ProPath templates to document them. Additional test guidance from ProPath and Harris process documents help to guide the creation of the test cases/scripts. Harris utilizes the Requirements Traceability Matrix (RTM) to map test cases/scripts to the RSD specifications.

## Defect Reporting and Tracking

This section describes how defects are defined, the severity levels, and the defect resolution process.

Defects found during testing will be recorded in the Harris SharePoint Test Tracker. Beta testers will use an Excel log to record defects. The log will be imported to the Harris SharePoint Test Tracker and VA AWARE SharePoint.

### Defect Specification

Test cases for each iteration must conform to a step/verification point format, where a particular action is defined and an immediate verification of the expected response is noted. At each verification point, testers make a pass/fail decision. Execution of a test case is only considered successful if every verification point is deemed successful (i.e., the expected response is observed).

Each test case/test script defines the expected results for each test. When a test result does not match the expected result, the test fails and the impact of the test failure is determined.

When the test team discovers a defect, they assign a severity level to the defect based on the potential impact of the failure on the system. The Development Team and Project Management then determine the priority level of the resolution of that defect.

### Severity Definitions

When a failure occurs in testing, testers write a defect report against the test step where they observe the failure. Testers assign a severity level based on their observations when the failure occurs. The severity level of a defect may change, as it is analyzed and resolved. The four severity levels are

* Critical
* High
* Medium
* Low

**Critical**A critical defect is the highest level of defect. A critical defect prevents the test case from being run any further. Often, a critical defect causes all testing to stop. Attributes associated with a critical defect are:

* No acceptable workaround exists
* A system crash and/or data corruption
* An immediate resolution is needed

**High**A high defect is the second highest level of defect. A high defect often has a workaround that allows the test case to continue or allows other test cases to be run. Attributes associated with a high defect are:

* Significant negative impact to testing item
* Workarounds can be difficult to achieve
* The current test case may need to be stopped
* The application cannot proceed into production without a resolution

**Medium**A medium defect is the normal level of defect. A medium defect generally has a workaround and allows the test case to continue to run. Attributes associated with an average defect are:

* Impacts to the users of the applications
* An average value or loss
* Desirable to be fixed prior to the application proceeding into production

**Low**A low defect is an error or lack of functionality that may cause operator/user inconvenience and minimally affects operational processing. Developers can address this defect after the system is in production. An example of a minor defect is a typographical error.

### Defect Resolution Process

Defects found by the Test Team are reported to the Project Manager. They then determine a priority level that indicates the impact a defect has on AWARE. The priority levels are as follows:

* High
* Medium
* Low
* Future Release

**High**A highpriority level indicates that the impact of a defect is very significant. Some of the reasons for determining this level include:

* Adversely affects all users
* Represents a significant value or loss
* Represents a major security failure
* Does not allow an acceptable workaround
* System crash and/or data corruption
* An immediate resolution is needed

**Medium**A mediumpriority level indicates that the impact is significant. Some of the reasons for determining this level include:

* Inconvenience; workaround is difficult to achieve
* Component(s) malfunction, which creates major functional or logical application errors
* Must be fixed prior to the application proceeding into production

**Low**A lowpriority level indicates that the impact is not significant. Often a low priority level can be changed to future (to be resolved in a future release) or no problem (which means it does not need to be fixed). Some of the reasons for determining a low priority include:

* Inconvenience, but allows an acceptable workaround
* Does not create major functional or logical application errors
* Desirable to be fixed prior to the application going into production

**Future Release**A future releasepriority level indicates that the impact is not significant or is tied to changes coming in a future release.

## Metrics and Reporting

Metrics are a system of parameters or methods for quantitative and periodic assessment of a process. The Rational® suite of test tools provides a number of metrics. Metrics reports will be generated as needed.

Some of the test metrics available are:

* Number of test cases that passed or failed
* 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

## Testing Tools

The Testing tools employed to support the project testing are listed in Table 3 below.

Table 3 – Tool Category or Types

| Tool Category or Type | Tool Brand Name | Vendor or In-house | Version |
| --- | --- | --- | --- |
| Test Management | Excel | Microsoft |  |
| Defect Tracking | SharePoint | Microsoft |  |
| Configuration Management | Tortoise Subversion (SVN) | GNU General Public License |  |
| Project Management | Microsoft Project | Microsoft |  |
| Requirements Traceability | Excel | Microsoft |  |
| 508 Accessibility Testing | FireEyes  WAVE | Worldspace  WebAIM |  |

## Testing Risks and Mitigation

The Testing Risks are as follows:

* Testing materials are inaccurate and may result in false results
* Testing resources are not familiar with the product being tested

The Mitigations are:

* Testing materials will be reviewed and approved by VA
* VA will identify Subject Matter Experts to perform the testing

## Product Component Testing

The AWARE Component Test is a continual, informal test performed during development and beta readiness. The component test isolates each part of the application to show that the individual parts are correct. These tests are executed in the Development Team’s Innovation Sandbox Development environment.

Product Component tests are the internal technical and functional testing of a module or component of code. AWARE developers conduct these tests. It is the responsibility of each individual developer to test their code module properly in the developers’ environment. The source code will be tested and provided within the scheduled time outlined in the AWARE project schedule. Steps include:

* Identifying the areas that have changed during the current development phase
* Executing tests to verify that the changes for the current development phase
* Performing exploratory testing i.e., randomly exercise the module, object, and options based upon domain knowledge, past performance, and expertise
* Code fixes and retesting

## Component Integration Testing

Component integration testing is performed after development in the Innovations sandbox. The Development Team performs this testing to verify that the functionality introduced in the development addresses the associated requirement(s), the integrated website continues to function as expected, and links to external resources function as expected. The intent is to use many of the tests developed for component integration testing for system testing as well.

Product component integration testing is performed to expose defects in the interfaces and interaction between integrated components. Any defects found are resolved and The Test Team re-tests the affected component. If a defect cannot be resolved, the Development Team puts the portion of code containing the defect into the next sprint. When all tests are completed without failure, and any uncovered defects have been resolved, that sprint’s component testing is complete. The test results are reviewed at the sprint’s retrospective meeting.

The AWARE component integration tests verify that the functional requirements associated with the current sprint are successfully integrated with prior sprints’ output. These tests are executed at the end of the sprint after product component test conduct for that sprint. Activities include:

* Verifying the requirements associated with the current development phase
* Executing test scripts to verify each functional requirement
* Tracking identified defects
* Updating test scripts to reflect the website design, so that listed actions and expected results are accurate and current
* Executing tests with various combinations of options and data (e.g. test with minimal data entered and with maximum data entered)
* Performing exploratory testing (i.e., randomly exercise the module, object, and options based upon domain knowledge, past performance, and expertise)

## System Tests

System Testing is the level of testing done to verify that the various components of a system interact and pass data correctly among one another and function cohesively. Test scripts will be developed to test that all components interact correctly. The integration tests are built based upon the design specifications. Test cases will include:

* Positive testing: A test of the system’s functional and structural integrity to determine that it performs as intended under normal conditions
* Boundary testing: A test of input fields using values that lie at either extreme of the valid input range
* Negative testing: A test of input errors using invalid values
* Testing will be done in the VA Innovations Sandbox Environment

## User Functionality Test (UFT)

With the successful completion of the AWARE Beta Readiness Demonstration and installation in the VistA/CPRS test account environment, VA personal will evaluate usability and performance at the beta and pilot test sites within the test account environment. The purpose of UFT is to provide confirmation that the business requirements are properly and fully addressed in AWARE.

The VistA/CPRS Test Environments will be determined by the VA Program Manager.

## Testing Service Test

Not applicable to AWARE.

## Operational Readiness Testing Support

Not applicable to AWARE.

## Initial Operating Capability Testing

Not applicable to AWARE.

# Testing Techniques

This section describes the approach to risk-based testing, requirements for enterprise testing, test types, iterations, and tools used to test the test items.

## Risk-Based Testing

Risk-based testing is a technique for prioritizing testing based on testing the highest risk items first and continuing down the risk prioritization ladder as the testing schedule permits.

## Enterprise Testing

This section describes how the project testing covers the enterprise requirements. Enterprise requirements include the following: security, privacy, Section 508 compliance requirements, and multi-divisional requirements.

### Security Testing

Security testing validates the requirements specified in “Security Specifications” in the project Requirements Specifications Document (RSD). The Test scripts will be created by the test team to verify readiness for independent testing performed by the VA Security Assessment Team as required by the Test and Certification Process.

### Privacy Testing

VA/Privacy team will verify:

* Veteran and employee data are adequately protected in AWARE
* The AWARE solution complies with the Privacy and Security Rule provisions in the Health Insurance Portability and Accountability Act (HIPAA)

### Section 508 Compliance Testing

Section 508 Compliance Testing is required for an application utilizing Graphical User Interface (GUI).

The Harris test team is responsible for verifying that product functionality is usable from the keyboard, while the Section 508 Program Office is responsible for performing independent compliance testing with assistive technology. This test type validates the requirements specified in “Accessibility Specifications” in the project RSD.

### Multi-Divisional Testing

Multi-Divisional Testing is required by VA to validate all applications will operate in a multi-division or multi-site environment, recognizing an enterprise perspective while fully supporting local health care delivery.

Testing will be performed to verify and validate that the product or system complies with the multi-divisional requirements as specified in “Multi-Divisional Specifications” in the RSD. Testing in a multi-divisional environment will occur in all of the major testing phases.

Component integration testing and system regression testing will be performed in a test environment that emulates a multi-divisional environment.

## Test Types

Discrete test types that are generally combined in a group’s testing effort are listed below in Table 4. Component integration testing and functional requirement testing are examples of different test types. Test cases may include aspects of both simultaneously.

Table 4 – Test Types to be Performed

| Test Type | Responsible Party |
| --- | --- |
| Product Component test | Development Team |
| Integration Testing | Development/Test Team |
| Installation Test | Development/Test Team |
| Regression Testing | Development/Test Team/User Works |
| Smoke Testing | Development/Test Team/User Works |
| System Testing | Development/Test Team/User Works |
| Section 508 Compliance Test | Section 508 Program Office |
| Beta/User Functionality Testing | VA Field Test Sites |

## Iterative Testing

For AWARE, iterative testing for Product Component Test and Component Integration Test will be performed for each sprint where components are developed or changed. After the sprints are completed, tests are performed in a non-iterative fashion. Re-testing occurs for areas of the code where defects are found and corrected by the Harris Development Team.

## Productivity and Support Tools

Please refer to Section 3.7, Testing Tools.

# Test Criteria

## Process Reviews

This MTP underwent a peer review prior to final approval by the Harris team and submittal to VA.

## Pass/Fail Criteria

Test cases must conform to a step/verification point format, where a particular action is defined and an immediate verification of the expected response is noted. At each verification point, a pass/fail decision is made. Execution of a test case is only successful if every verification point passes successfully.

Each test case/script defines the expected results for each test. When a test result does not match the expected result, the test fails and the impact of the test failure is assigned. When the Test Team discovers a defect, they assign a severity level to the defect based upon the potential impact of the failure on the system. Project Management and the Development Team then determine the priority level of the defect’s resolution.

## Suspension and Resumption Criteria

The suspension of testing occurs when testers encounter a critical or high defect. The error may apply to a particular area of the product and often causes all testing to be suspended. When a critical or high defect occurs, the management team will discuss the impact of the defect and determine if testing needs to be suspended and which product areas are affected. The test suspension causes testing to be suspended or blocked until a fix or workaround can be installed and unit tested.

Suspension conditions include:

* A defect discovered during test execution blocks all further progress until the defect is resolved
* The test environment is corrupted or rendered unusable
* Project management determines the need to suspend testing based on some other criteria

Resumption criteria are:

* The condition that caused the suspension is addressed
* The changes are component tested
* The changes are applied to the test environment
* The initial entry criteria of the test phase are met
* For example, meeting the initial entry criteria for a resumption of UFT testing includes component and component integration re-testing internally

## Acceptance Criteria

Acceptance criteria are decision rules that a component or system must satisfy in order to be accepted by a user, customer, or other authorized entity. They include:

* The testable, functional requirements from development must be tested and verified
* All defects with a priority level of critical, high and medium are resolved, retested and closed
* All defects with a priority level of low are resolved or changed to a defect resolution level of no problem or future release
* Any outstanding test cases from a previous test run have successfully passed
* All the test cases are executed without defects at a medium or higher level
* Test Documentation is updated (See for details)

# Test Deliverables

Test deliverables for the AWARE project, and the party/person responsible for completing the deliverables are listed below in Table 5:

Table 5 – Test Deliverables

| Deliverable | Responsible Party |
| --- | --- |
| Requirements Traceability Matrix (RTM) | Harris Analyst Team |
| Test Cases | Harris Analyst Team |

# Test Schedule

A complete test schedule can be found in the AWARE SharePoint site.

# Test Environments

A test environment is an environment containing hardware, instrumentation, simulators, software tools, and other support elements needed to conduct a test.

The test environments will be hosted at VHA Innovations Sandbox-Lab and the Beta & Pilot Site Test accounts as determined by VA.

## Test Environment Configurations

In order for a successful test, there needs to be a controlled environment that resembles as close as possible the production environment for the application. Configuration Management (CM) must exist on the environment so each version of all associated applications is known during each test cycle. Any changes during a test cycle can affect the outcome of the individual test cases, and the overall test cycle and invalidate the test phase. The Development Team will be responsible for software builds, VistA Patch installs, setup, and configuration.

## Base System Hardware

The system resources for the test effort presented in this Master Test Plan are listed below in Table 6:

Table 6 – System Hardware Resources

| System Hardware Resources | | |
| --- | --- | --- |
| Resource | Quantity | Name and Type |
| Database/Application/Web Server | 1 | Dell Power Edge Server |
| Intersystems Cache Server | 1 | VistA Server |
| Test Repository | SharePoint | AWARE |

## Base Software Elements in the Test Environments

The software elements that are required in the test environment for this MTP are listed below in Table 7:

Table 7 – Software Elements

| Software Element Name | Version | Type and Other Notes |
| --- | --- | --- |
| MS Windows Server | 2008 | Operating System |
| MS SQL Server | 2012 | Database |
| MS SQL Reporting Services | 2012 | Reports |
| MS IIS (Internet Information Services) | 7.5 | Web Server |
| ASP.NET MVC3 | N/A | Web Application |
| Intersystems Cache Server | 20xx | Database |
| MS IIS CSP Gateway Module | Based on Cache Version | Web Application |

# Staffing and Training Needs

Staffing resources needed to plan, prepare, and execute this MTP are listed below in Table 9. Please refer to the AWARE Staff for a complete listing of staffing resource requirements.

Table 8 – Staffing Resources

| Testing Task | Quantity of Personnel Needed | Test Process | Duration/ Days |
| --- | --- | --- | --- |
| Create MTP | Per resourced schedule | Test Preparation | Per Resourced Schedule |
| Establish Test Environment | Per resourced schedule | Test Preparation | Per Resourced Schedule |
| Perform System Tests | Per resourced schedule | Product Build | Per Resourced Schedule |

# Constraints

The risks identified in this MTP are recorded in the Harris SharePoint Risk List.

The following actions are taken to prevent any adverse impacts upon the testing of critical program functionality:

* Identify test execution risks
* Record the risks in a tracking tool
* Assess the potential impacts and probability for each risk
* Plan mitigation for each risk
* Monitor the risks through closure

The following constraints that affect the testing activity have been identified:

* Unit Testing must have successfully been completed prior to System Testing
* The Test Environment must be available, tested and fully operational prior to the commencement of testing
* Data required for testing must be available and reusable

# Acronyms

Table 9 – Acronyms and Abbreviations

| Abbreviations, Terms | Definitions |
| --- | --- |
| AWARE | Alert Watch and Response Engine |
| CM | Configuration Manager/Management |
| CPRS | Computerized Patient Record System |
| GUI | Graphical User Interface |
| HeV-VistA | Healthy Veteran |
| HIPAA | Health Insurance Portability and Accountability Act |
| IIS | Internet Information Services |
| KIDS | Kernel Installation & Distribution System |
| MTP | Master Test Plan |
| OIT | Office of Information & Technology |
| PjM | Project Manager/Management |
| QI | Quality Improvement |
| PMAS | Project Management Accountability System |
| PWS | Performance Work Statement |
| RSD | Requirements Specification Document |
| RTM | Requirements Traceability Matrix |
| SQL | Structured Query Language |
| T4 | Transformation Twenty-One Total Technology |
| UAT | User Acceptance Test |
| UFT | User Functionality Test (previously called UAT) |
| UI | User Interface |
| VA | Department of Veterans Affairs |
| VACI | Department of Veterans Affairs Center for Innovation |
| VHA | Veterans Health Administration |
| VistA | Veterans Health Information Systems and Technology Architecture |

# Test Types Descriptions (and Minimum Set of Test Types Required)

Test types are test techniques used to validate the system or application. This table lists possible test types that may be utilized during the Product Build, Testing Service Testing, and Beta Testing.

The Test Team, in consultation with the Development Manager, selects the test type(s) best suited to the system or application being tested. The “X” mark shows that a particular testing type is recommended by Harris standards, and which phase it will occur within (see the three “Testing” column labels).

The “◊” symbol appears in conjunction with the “X” mark. The “◊” symbol designates that a testing type is part of an overall minimum set of test types required.

Tests other than those listed here may be added at the discretion of the Project Team.

Table 10 - Test Types

| Test Types | Description | Product Build Testing◊ | System Testing◊ | Beta Testing◊ |
| --- | --- | --- | --- | --- |
| Access Control Test | Verifies access to the system is controlled and that unwanted or unauthorized access is prohibited. 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. | X**◊** |  |  |
| Build Verification Test | Testing performed for each new build, comparing the baseline with the actual object properties in the current build. |  | X**◊** |  |
| Business Cycle Test | Verifies the functionality associated with a period of time (e.g., daily, monthly, annually), focusing upon activities and transactions performed end-to-end over time. |  |  | X**◊** |
| Component Integration Test | Testing performed to expose defects in the interfaces and interaction between integrated components as well as verifying installation instructions. | X**◊** | X**◊** |  |
| Configuration Test | Verifies the program is compatible with the different configurations of hardware and system software as defined in the requirements. The goal of the configuration test is finding a hardware combination that should be, but is not, compatible with the program. |  |  |  |
| Contention Test | Tests executed 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 Test | Testing 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 due to data’s initial storage, updating, restoration, or retrieval processing. | X**◊** |  |  |
| Documentation Test | Verifies all the required information (software documentation set) 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. | X**◊** |  | X**◊** |
| Error Analysis Test | Verifies 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. |  |  | X**◊** |
| Exploratory Test | 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. | X**◊** | X**◊** | X**◊** |
| Failover Test | Verifies 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 is usually performed in combination with Recovery Testing. |  |  |  |
| Installation Test | Verifies 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) as well as testing the uninstall process (returning the system, application and database to the state prior to the install). | X**◊** |  |  |
| Integration Test | Verifies the combination 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). | X**◊** | X**◊** |  |
| Load Test | Verifies 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 Test**◊** | 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 Test**◊** | Verifies all applications operate in a multi-division or multi-site environment recognizing that an enterprise perspective while fully supporting local health care delivery. |  |  |  |
| Parallel Test | 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 Test | 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. |  |  |  |
| Privacy Test**◊** | A type of testing that verifies 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). | X**◊** |  |  |
| Product Component Test  (Unit Testing) | Verifies the requirements defined in the detail design specification have been successfully applied to the module/component under test. | X**◊** |  |  |
| Recovery Test | Verifies recovery processes are invoked and data recovery is achieved while an application or system is monitored during a controlled application or system failure. Recovery Test is usually combined with Failover Testing. |  |  |  |
| Regression Test | Verifies existing functionality still performs as expected when new functionality is introduced into the system under test. | X**◊** |  |  |
| Risk-Based Test | Testing designed to explore and/or uncover potential system failures by using the list of risks to select and prioritize testing. |  |  |  |
| Section 508 Compliance Test**◊** | A type of test that (1) verifies 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. | X**◊** |  |  |
| Security Test**◊** | Verifies the security requirements and readiness for independent testing performed by the Security Assessment Team (as required by the Test and Certification Process). | X**◊** |  |  |
| Smoke Test | Verifies the 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. | X**◊** | X**◊** |  |
| Stress Test | 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 Test | Verifies the system performance, operation, and functionality are sound. End-to-end testing with all interfacing systems is the ultimate goal. | X | X**◊** |  |
| Usability Test | 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. | X**◊** | X**◊** | X**◊** |
| User Functionality / Acceptance Test (UAT) | Testing that involves end-users validating the functionality of the application using test data in a controlled test environment. |  |  | X**◊** |
| User Interface (UI) Test | Testing the user interfaces to verify the interfaces follow accepted standards and meets requirements. | X**◊** | X**◊** | X**◊** |

# Attachment A - Approval Signatures

The following members of the governing IPT are required to sign. Please annotate signature blocks accordingly.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Signed: Date:  
Blake Henderson   
Project Manager  
Innovation Coordinator

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Signed: Date:  
Brian Stevenson   
Contracting Officer’s Representative  
Innovation Coordinator  
VHA OIA Innovation