**Department of Veterans Affairs (VA):**

**Office of Information & Technology (OI&T)**

**Product Development**

**Memorial Benefits Management System (MBMS)**

**Memorial Cemetery Management Modernization – Phase 1 (CM1)**

**Remains Tracking Test Plan**

****

**April 2016**

**Version 0.4**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Author |
| 04/14/2016 | 0.4 | Updated with COR’s comments |  |
| 02/10/2016 | 0.3 | Initial Updates |  |
| 02/03/2016 | 0.2 | Initial Draft Updates |  |
| 01/12/2016 | 0.1 | Initial Draft |  |

**Table of Contents**

[1. Introduction 1](#_Toc448422366)

[1.1. Purpose 1](#_Toc448422367)

[1.2. Scope 2](#_Toc448422368)

[1.3. Test Objectives 2](#_Toc448422369)

[1.4. Roles and Responsibilities 2](#_Toc448422370)

[1.5. Reference Documents 3](#_Toc448422371)

[2. Items to Be Tested 3](#_Toc448422372)

[2.1. Access, Save and Submit Workflows on a Handheld Device 3](#_Toc448422373)

[2.2. Manage Access to the Enterprise System 3](#_Toc448422374)

[2.3. Manage Remains Tracking Workflow 4](#_Toc448422375)

[3. Testing Environment 4](#_Toc448422376)

[3.1. Software Elements in the Test Environments 4](#_Toc448422377)

[4. Test Approach 5](#_Toc448422378)

[5. Types of Testing 5](#_Toc448422379)

[5.1. Unit Testing 5](#_Toc448422380)

[5.2. Integration Testing 5](#_Toc448422381)

[5.3. System Testing 5](#_Toc448422382)

[5.4. Regression Testing 6](#_Toc448422383)

[5.5. 508 Compliance Testing 6](#_Toc448422384)

[5.6. User Acceptance Testing 6](#_Toc448422385)

[5.7. System Capacity Availability and Performance 6](#_Toc448422386)

[6. Testing Process 6](#_Toc448422387)

[6.1. Test Procedure 7](#_Toc448422388)

[6.2. Defect Resolution 7](#_Toc448422389)

[7. Testing Criteria 7](#_Toc448422390)

[7.1. Entry Criteria 7](#_Toc448422391)

[7.2. Exit Criteria 8](#_Toc448422392)

[7.3. Pass/Fail Criteria 8](#_Toc448422393)

[7.4. Acceptance Criteria 9](#_Toc448422394)

[8. Reporting/Test Metrics 9](#_Toc448422395)

[9. Testing Tools 9](#_Toc448422396)

[10. Test Deliverables 10](#_Toc448422397)

[11. Test Schedule 10](#_Toc448422398)

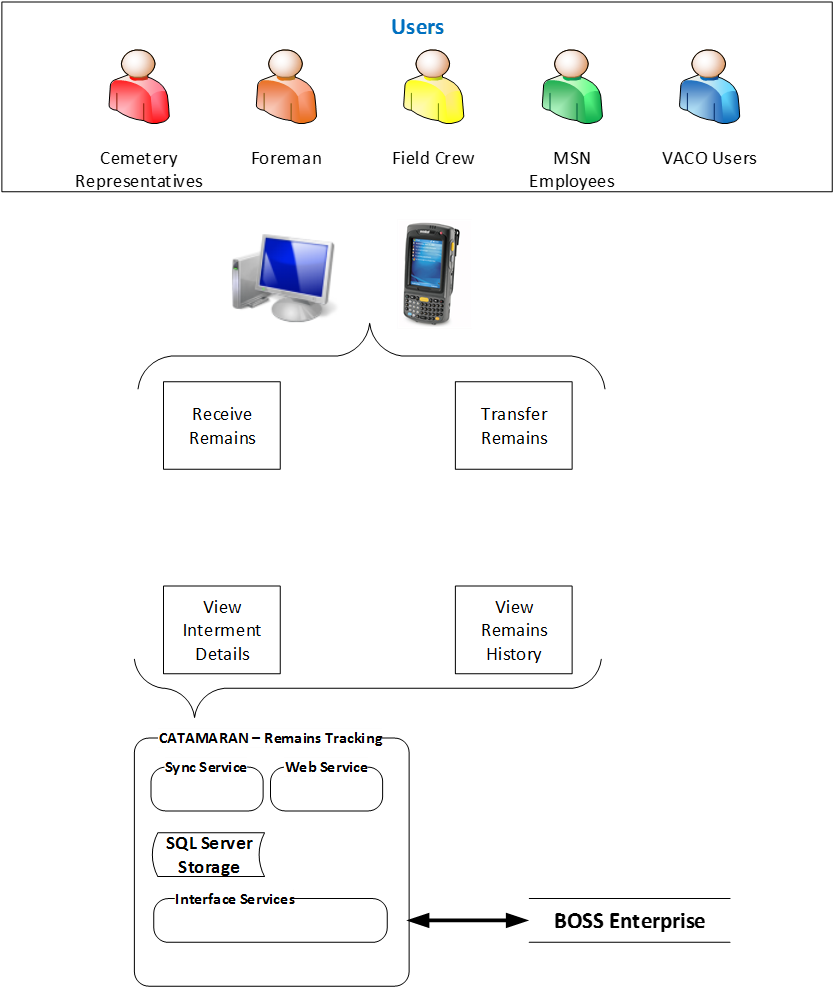
[12. Test Type Definitions 11](#_Toc448422399)

# Introduction

## Purpose

The purpose of this test plan is to analyze, define and execute testing activities for software development compiled for the COTS (Commercial-off-the-shelf) product CATAMARAN, which provides burial and memorial serves to Veterans and eligible family members in an effective, timely and compassionate manner. This plan’s main objective is to validate The Remains Tracking workflow functionality from the Funeral Home/Family to Cemetery Representatives to the Foreman/Caretaker to interment. The casket, urn, temporary marker, gravesite, and permanent marker will be verified and identified using the technological solution Automated Identification and Data Capture (AIDC). Handheld devices will be used to implement and validate this solution. The accountability of remains begins at the time the remains enter the cemetery through interment at national cemeteries.

**Remains Tracking Application Overview**



**Figure 1: Remains Tracking Application Context Diagram**

## Scope

This Plan defines the testing details that will be executed for the stakeholders of the Memorial Cemetery Management Modernization – Phase 1 (CM1) project. This plan details the system test processes and procedures from pre-test preparation, test readiness reviews, exit criteria, and reporting. It will provide an overview of testing activities of the management system for the standardized solution to be created for the CM1 project that will enable National Cemetery Administration (NCA) to perform the critical tasks to accurately track remains in a stable system. The system testing process provides detailed testing activities that will ensure that the software developed is bug free and functionally stable. This document represents the envisioned application for CM1 (Increment 1 Remains Tracking) as documented in the NCA Workflow Business Requirements Document (BRD), April 2015, Version 1.0.and Requirement Specifications Document (RSD).

## Test Objectives

This Test Plan supports the following objectives:

* To provide test coverage for 100% of the documented requirements (Increment 1 Remains Tracking)
* To provide coverage for System/Software Design Document elements
* To execute 100% of the test cases during User Functionality Testing
* To create, maintain, and control the test environment

## Roles and Responsibilities

List of the key roles and their responsibilities for this Test Plan

Table 1: Roles and Responsibilities

| Role | Description |
| --- | --- |
| Project Manager | Monitor execution of testing activities. |
| Development Team | Persons that build or construct the product/product component and Perform dynamic software testing (Unit testing) |
| QA (Quality assurance) / Test Manager | Manages development of test plans, test case design, formal testing of the system.  Manages coordination of the system test environment.  Establishes the methodology for conducting testing.  Sets testing milestones/schedules.  Identifies and reports testing related risks and recommends risk mitigation strategies. |
| Test Engineer | Develops and maintains the Test Plan.  Defines, sets up, and executes the verification scripts.  Test reports and Test analysis. |
| Test Environment Team | Persons that establish, maintain, and control test environments. |

## Reference Documents

This section identifies the documents referenced during the creation of this document.

* Contract Project Management Plan (CPMP)
* System Design Document (SDD)
* Requirements Specifications Document (RSD)
* Requirements Traceability Matrix (RTM)

# Items to Be Tested

The items described below will be tested during the Test and Development phases of CM1:

## Access, Save and Submit Workflows on a Handheld Device

* Provide access to a portable handheld device.
* Provide for updates from a handheld device.
* Provide for the ability to synchronize recorded workflow information captured from the handheld device when connectivity is available.

## Manage Access to the Enterprise System

* Provide a view of the workflow from within the enterprise system.
* Provide update capability of the workflow from within the enterprise system.
* Provide the capability to add new workflows.

## Manage Remains Tracking Workflow

* Provide for the ability to support Automated Identification and Data Capture (AIDC).
* Assign a unique AIDC number to a decedent at the time eligibility is established.
* Allow for the inclusion of the AIDC number on tags, documents, or system screens that pertain to the interment.
* Allow for retrieval of information from the portable device using the AIDC number.
* Allow for identification of interment types.
* Allow for multiple transfers of remains.
* Allow for acceptance of remains
* Track remains from arrival on cemetery grounds to interment

# Testing Environment

The test environment contains hardware, instrumentation, simulators, software tools, and other support elements needed to conduct a test. Testing will be performed across multiple environments, as shown below.

The CM1 system will have 3 environments:

* Development Environment, used by the development team
* Testing environment, used by the QA and UAT (user acceptance testing) teams
* Production environment

## Software Elements in the Test Environments

Describes the base software elements that are required in the test environment for this Test Plan.

| **Software Element Name** | **Version** | **Type and Other Notes** |
| --- | --- | --- |
| Workstation | Windows 7/ Pro | Operating System |
| Microsoft Windows | Windows 7 /Pro (NT601) | Operating System |
| Internet Explorer | IE 8,9,10,11 | Internet Browser |
| Network Associates McAfee Virus Checker | Latest | Virus Detection and Recovery Software |

# Test Approach

ProSphere will provide an integrated and complete end-to-end functional testing solution to developed software with its defined agile methodology. A well-defined end-to-end metric based process with proper entry and exit criteria focusing on the entire aspects of performance testing is followed. The approach to be followed is based on functional, security and role based objectives. The Test Approach cites how the Development Team plans to cover the testing activities specified in the Product Build and Independent Test and Evaluation processes in ProPath.

# Types of Testing

The following types of testing will be conducted on the CM1 project

## Unit Testing

Unit testing is the internal technical and functional testing of a module of code. The developer conducts this process using automated unit and functional testing with Continuous Integration framework.

## Integration Testing

Integration testing is an incremental series of tests of combinations or sub-assemblies of selected components in an overall system. This test will ensure interoperability of components, internal and external, and will ensure interface compliance and integrity of CM1. The ProSphere test team will conduct integration testing of the core capabilities of the application flow. This testing will be based on detailed business and user requirements.

## System Testing

System testing involves scenario based functional tests that are written from a user's perspective. These tests confirm that the system does what users are expecting it to do, and are performed without consideration to the underlying product architecture or composition. System testing will be performed by the ProSphere test team and will be conducted in the development test environment (DTE) located in Austin, TX (AITC). The test team will create functional test scripts for the purposes of system testing. All test scripts will be maintained in the Rational Quality Manager (RQM) tool. Traceability will be maintained as described in the Requirements Traceability Matrix (RTM)

## Regression Testing

Regression testing will be executed to verify that modifications have not caused unintended adverse side effects, while also ensuring new features or fixes to previously reported defects are successful. The regression test scripts will include testing of all services in CM1, end to end, and interface scenarios that are not subject to change. A round of regression testing will be conducted after new functionality has been verified during the system testing phase of each release cycle. Regression testing efforts will target areas identified by the Test Manager as critical, high-risk, and/or known code change. The ProSphere team will conduct regression testing for each release of CM1.

## 508 Compliance Testing

Section 508 requires that all Web site content be equally accessible to people with disabilities. This applies to Web applications, Web pages and all attached files on the intranet, as well as, internet.

508 Compliance Testing will be performed as specified in the CM1 PWS and RTM.

## User Acceptance Testing

The CM1 Product Owner will conduct user acceptance and functional testing with the use of test service endpoints or simulators and will conduct evaluation efforts in the Pre-Production environments. The ProSphere testing team will provide support as needed.

## System Capacity Availability and Performance

In accordance with NCA standards, the CM1 application capacity will function seamlessly with 400 users 6 days per week and 13 hours per day. System would perform > 10 transaction per user in each hour. The anticipated peak user time is Monday through Friday (10:00 AM to 3:00 PM EST). NCA expect to add approx. 260 users on first year and around 285,000 transactions per year will be added. System would have a search capacity of (11-1000 per hour). The System should be available for 99.9% of time to users on 24x7 basis on all time zones. Acceptable down time is 8.76 hours yearly.

# Testing Process

This section outlines the verification elements such as Verification Test Cases, and models that will be used to construct the overall verification.

## Test Procedure

The test team will test each step within the test procedure, validating that the system works as designed and that requirements have been met. Test cases will capture the following types of data:

* Record-specific input data
* Record data that will be used for further test analysis
* Actual results of testing, especially if different than the expected test results
* Problems associated with a specific test step will be dealt with according to the Configuration Management processes

A test case passes if the actual result matches the expected result. If the actual result does not match the expected result, it is treated as a failed test case. If a test case fails, it is not assumed that the feature is defective. For example, misinterpretation of project documentation, incomplete documentation, or inaccurate documentation can cause failures. Each failure is analyzed to discover its cause, based on actual results and the results described in project documentation.

Refer to the CM1 test cases in RQM for detailed information.

## Defect Resolution

When testing is complete and defects identified, the defects will be documented and severity will be assigned according to the severity ratings in Table 4.

Table 4: Defect Resolution Table

|  |  |
| --- | --- |
| Severity Rating | Description |
| Critical | Requires an immediate fix; blocks further testing; Highly visible; Results in system crash, data loss, or corruption |
| High | Requires a fix before the system is released; Results in an operational error, wrong result, or loss of functionality |
| Medium | Requires a work-around but the system still functions and is useable; Requires a fix before the system is released |
| Low | Should be fixed if time permits; Results in a minor problem such as misspellings and is a rare occurrence |

The development manager monitors the defects encountered during testing. Defects are then assigned to developers based on severity and will be documented in the Defect Tracking Log

# Testing Criteria

## Entry Criteria

All entry criteria must be met before system testing can begin.

Entry Criteria is as follows:

* Development code completed
* Successful execution of unit tests
* Completed Release Cycle Plan
* 100% test case coverage of requirements
* Traceability of requirements
* Sub-system design specifications are complete
* Development team has handed over software for installation in the testing environment(s)
* Smoke test (sanity test) is completed and passed
* Informal walkthrough completed

## Exit Criteria

Exit criteria is met when:

1. All the tests have been completed before the time given for UAT, or
2. The period of time to complete all the tests has ended, or
3. The test team discovers too many defects to continue testing, or
4. The test team discovers a defect that prevents the continuation of any additional testing.

Testing is complete when:

* One of the exit criteria is met, and
* All defects identified have been reported and logged into the defect tracking system, and
* A final Test Report has been generated, providing recommendation on whether the product should go to UAT.

## Pass/Fail Criteria

The following definitions are used when dealing with test results:

* + **Pass:** The capability tested operates in a way usable for the business and meet the stated requirements in the CM1 RTM (Increment 1 Remains Tracking)
  + **Pass with exceptions:** The capability tested operates mostly according to the business needs, meet the business requirements, but suffers from minor defects. However, these defects do not stop the business from operating, and do not require workarounds to be created.
  + **Fail:** The capability tested does not meet the requirements, or does not allow the business to operate according to its established guidelines. Workaround may (major defect) or may not (critical defect) be available
  + **Block:** The capability tested does not allow the business to complete its operation (critical defect)

## Acceptance Criteria

As a rule of thumb, a release is considered as "passed" when:

* + No critical defects (level 1) are found, and
  + No major defects (level 2) are found
  + 7 Medium (level 3) defects (Minimum acceptable defect threshold)
  + 12 Minor (level 4) defects (Minimum acceptable defect threshold)
  + The Test report is created and delivered to leadership.

# Reporting/Test Metrics

Reporting or metrics is a system of parameters or methods for quantitative and periodic assessment of a process to be measured.

CM1 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

# Testing Tools

Describes the tools that will be used to support the testing activities

Table 5: Testing Tools

| Tool Category or Type | Tool Brand Name | Vendor or In-house | Version |
| --- | --- | --- | --- |
| Jazz Technology Platform | Rational Team Concert (RTC) | IBM |  |
| Jazz Technology Platform | Rational DOORS Next Generation (RDNG) | IBM |  |
| Jazz Technology Platform | Rational Quality Manager (RQM) | IBM |  |

# Test Deliverables

| **Test Deliverables** | **Responsible Party** |
| --- | --- |
| Remains Tracking Test Plan | ProSphere QA / Test Team |
| Remains Tracking Functional Test Scripts | ProSphere QA / Test Team |
| Remains Tracking Test Results | ProSphere QA / Test Team |
| Test Data | ProSphere QA / Test Team / ProSphere Development Team |
| Test Environment | ProSphere QA / Test Team / ProSphere Development Team |
| Traceability Report or Matrix | ProSphere QA / Test Team |

# Test Schedule

| **Testing Milestones** | **Scheduled** | **Responsible Party** |
| --- | --- | --- |
| Remains Tracking Test Plan | 04/26/2016 | ProSphere QA / Test Team |
| Remains Tracking Functional Test Cases/Test Scripts | 05/30/2016 | ProSphere QA / Test Team |
| Test Environment Readiness  (CATAMARAN Server) | 03/01/2016 (will move from ProSphere environment to VA environment after Elevated Privileges complete) | ProSphere Test & Development Team |
| Build Process | 05/27/2016 | ProSphere Development Team |
| Complete Test Execution | 06/10/2016 | ProSphere QA / Test Team |
| User Acceptance Test (UAT) | 06/13/2016 | VA Team |
| Performance Test | 06/10/2016 | ProSphere (CATAMARAN) Team |
| Remains Tracking Test Results | 06/30/2016 | ProSphere QA / Test Team |
| Traceability Report or Matrix | 01/29/2016 and as requested | ProSphere QA / Test Team |

# 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. |
| 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. |
| 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 re-hosting - 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. |
| 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. |
| 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. |
| 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). |
| 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. |
| 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. |
| Regression Test | A type of testing that validates existing functionality still performs as expected when new functionality is introduced into the system under test. |
| 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 required 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 Acceptance Test | User Functionality Test (UAT) 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. |