Master Test Plan

Genomic Information System for Integrated Science 2

(Genisis2) Technical Services

Release 3



June 2017

Document Version 2.0

Department of Veterans Affairs

Document Revision History

| Date | Revision | Description | Author |
| --- | --- | --- | --- |
| 11/23/2016 | Draft | Initial draft of Master Test Plan | Booz Allen Hamilton |
| 11/25/2016 | 1.0 | Updates based on review by RTA | Booz Allen Hamilton |
| 11/27/2016 | 1.1 | Updates based on review by NK | Booz Allen Hamilton |
| 11/30/2016 | 1.2 | Updates to Section 8: Test Environments, ESE Testing and removal of footer from title page. | Booz Allen Hamilton |
| 01/25/2017 | 1.3 | Updated to Section 3: Test Approach and deletion of Section 3.7: Initial Operating Capability Evaluation | Booz Allen Hamilton |
| 04/10/2017 | 1.4 Draft | Updated the following sections for Release 2:   * 2.1 Test Inclusions * 3.0 Test Approach (Release 2) * 8.3 AITC (Test Environment Configuration)   Draft version until Release 2 is approved by the Business Owners | Booz Allen Hamilton |
| 04/17/2017 | 1.4 Draft | Final updates for Release 2 that includes updates to the following sections:   * 2.1 Test Inclusions * 3.0 Test Approach (Release 2) * 8.3 AITC (Test Environment Configuration) | Booz Allen Hamilton |
| 05/12/2017 | 2.0 | Updated Release 2 Functionality (pg.9) | Booz Allen Hamilton |
| 06/16/2017 | 3.0 | Updated the following sections for Release 3 functionality:   * 2.1 Overview of Test Inclusions * 3.0 Test Approach (Release 3 Features and Functionality) * 4.2 Enterprise Testing * 7.0 Test Schedule * 8. Staffing and Training Needs * 10. Risks and Constraints | Booz Allen Hamilton |

Table of Contents

[1. Introduction 1](#_Toc482264595)

[1.1. Purpose 1](#_Toc482264596)

[1.2. Test Objectives 2](#_Toc482264597)

[1.3. Roles and Responsibilities 3](#_Toc482264598)

[1.4. Processes and References 5](#_Toc482264599)

[2. Items to Be Tested 6](#_Toc482264600)

[2.1. Overview of Test Inclusions 6](#_Toc482264601)

[2.2. Overview of Test Exclusions 7](#_Toc482264602)

[3. Test Approach 7](#_Toc482264603)

[3.1. Product Component Test 11](#_Toc482264604)

[3.2. Component Integration Test 11](#_Toc482264605)

[3.3. System Tests 12](#_Toc482264606)

[3.4. User Acceptance Testing 12](#_Toc482264607)

[3.5. Enterprise System Engineering Testing 12](#_Toc482264608)

[3.6. Performance Testing 12](#_Toc482264609)

[4. Testing Techniques 12](#_Toc482264610)

[4.1. Risk-based Testing 13](#_Toc482264611)

[4.2. Enterprise Testing 14](#_Toc482264612)

[4.2.1. Security Testing 14](#_Toc482264613)

[4.2.2. Privacy Testing 14](#_Toc482264614)

[4.2.3. Section 508 Compliance Testing 14](#_Toc482264615)

[4.2.4. Multi-Divisional Testing 15](#_Toc482264616)

[4.3. Performance and Capacity Testing 15](#_Toc482264617)

[4.4. Test Types 15](#_Toc482264618)

[4.5. Productivity and Support Tools 17](#_Toc482264619)

[5. Test Criteria 17](#_Toc482264620)

[5.1. Process Reviews 17](#_Toc482264621)

[5.2. Pass/Fail Criteria 18](#_Toc482264622)

[5.3. Suspension and Resumption Criteria 18](#_Toc482264623)

[6. Test Deliverables 18](#_Toc482264624)

[7. Test Schedule 20](#_Toc482264625)

[8. Test Environments 21](#_Toc482264626)

[8.1. Test Environment Configurations 22](#_Toc482264627)

[8.2. AWS Test Environment Configuration 22](#_Toc482264628)

[8.3. VA (AITC) Test Environment Configurations 23](#_Toc482264629)

[8.4. Base Software Elements in the Test Environments 24](#_Toc482264630)

[9. Staffing and Training Needs 24](#_Toc482264631)

[10. Risks and Constraints 25](#_Toc482264632)

[11. Test Metrics 26](#_Toc482264633)

[Attachment A: Approval Signatures 27](#_Toc482264634)

[Appendix A: Test Type Definitions 28](#_Toc482264635)

# Introduction

The Department of Veterans Affairs (VA) has established leadership in genomic medicine through the undertaking of a ground-breaking program called the Million Veteran Program (MVP). Launched in 2011, MVP invites users of the VA healthcare system nationwide to participate in a longitudinal study with the aim of better understanding the interrelation of genetic characteristics, behaviors and environmental factors, and Veteran’s health. Today, with over 560,000 participants and recruiting at 50 VA sites nationally, MVP is the largest genomic research database in the world. The MVP data consists of blood samples from consenting Veterans that is used to generate genomic data, data from questionnaires and the electronic health record data. This resource is made available to VA researchers and VA-approved affiliates to pursue genomic discoveries and validation studies that can lead to personalized healthcare for Veterans. Ultimately, over the long term, validated scientific findings will be returned to the Veteran and incorporated into their medical record to fulfill the potential of personalized medicine – delivery of optimal interventions to patients based on their biological characteristics.

The primary component of the project is the Genomic Information System for Integrated Science 2 (Genisis2) application. Genisis 1.0, implemented over a four-year period (2011-2015), features a series of modular applications to facilitate recruitment and enrollment of MVP participants, automating most study-related logistics including study enrollment, clinical study data capture, consent, blood sample tracking, and genomic data storage. Genisis 1.0 also provides the secure analytical infrastructure necessary to conduct robust genomic and bioinformatics-related data management and data analysis. This remotely accessible analysis environment features a high-performance computing cluster with significant storage capacity and tools for scientific analysis of combined genotypic and phenotypic data.

Genisis2 will provide additional functionality to the Genisis 1.0 platform by automating data processing, data request transactions, and data request tracking for integrating VA Informatics and Computing Infrastructure (VINCI) clinical data into Genisis2, as well as enhanced systems administration capabilities.

The Master Test Plan is a living document that will be re-evaluated for each release of the Genisis2 application to ensure all aspects are adequately tested and implemented successfully. This document describes the test scope, environment, approach, and test resources involved in testing Genisis2.

## Purpose

The purpose of this document is to provide VA with a plan for test planning, documenting and execution of the Genisis2 releases in compliance with VA’s Office of Information and Technology (OI&T) Veteran-focused Integration Process (VIP). VIP is Lean-Agile framework that enables frequent releases of functional capabilities to be deployed that services the interest of Veterans through the efficient streamlining of activities that occur within the enterprise.

The Master Test Plan creates a functional roadmap to test each software build and planned release throughout the period of performance. It outlines the approach and tools that the Genisis2 Test Team will use to plan and test Genisis2, ensuring the system meets the functional, operational and compliance requirements as identified by the system and OI&T Compliance Epics.

The components of this Master Test Plan, along with the test deliverables, are maintained within Rational Requirements Quality Manager (RQM) in compliance with the VA’s VIP Methodology. This document will mirror the data elements housed within the Genisis Rational Instance and serve as a reference for those who do not have access.

This Master Test Plan will also support the following:

* Provide a central artifact to govern the planning and control of the test effort for Genisis2.
* Define the general testing approach that Genisis2 will follow.
* Demonstrate to Genisis2 stakeholders that various aspects governing the testing effort have been adequately considered; and where appropriate, have those stakeholders approve the plan.

This document will also support the following objectives:

* Identify sources of 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, types of tests, activities, and tools.
* Include the roles and responsibilities of the resources participating in the testing.
* Document the pass/fail performance of the software design parameters.

## Test Objectives

The Master Test Plan supports the following objectives:

* Outline the framework of testing at the program level and create a central artifact to govern the planning and control of the Genesis2 testing effort.
* Define stakeholders for testing across the project lifecycle.
* Define the processes, sequence, and test schedule with regard to high-level milestones and allocation of resources for management of the test phases across the project lifecycle that support:
  + Providing a comprehensive and consistent view of the test activities, work products, resources; and execution, environment, and integration efforts.
  + Identifying required test artifacts for each of the test phases.
  + Generating the test artifacts for the test phases.
  + Identifying and describing overarching processes critical to test execution, including entrance and exit criteria for test phases and milestones, and quality assurance.
  + Identifying and generating required reports for project status of the test phases.
  + Defining metrics for gauging progress in test execution and completion of verification of product development, product integration, product release, and site or field deployment of products.
  + Identifying risks inherent in the test execution and description of mitigation plans.
* Identify, create, maintain, and control the test environments.
* Execute 100% of the test cases during Integration, System, and Functionality Testing.
* Identify defect reporting processes.
* Identify and support additional needs for the project to meet VIP requirements or other OI&T needs (e.g., support for OI&T Compliance).

## Roles and Responsibilities

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

**Table 1: Roles and Descriptions**

| **Role** | **Description of Responsibilities** |
| --- | --- |
| **Program Project Manager** | * Overall responsibility for the successful planning and execution of a project. |
| **Development Project Manager** | * Provide general program/project coordination and oversight. * Provide sign-offs or identify resources for signing off on deliverables. * Review status of development and test efforts. * Provide input to the Master Test Plan. |
| **Development Scrum Master** | * Coordinates all development activities related to a project. * Review status of development and test efforts. * Provide input to the Master Test Plan. |
| **Business Functional Analyst** | * Create Requirements and Requirement Collections within RM for each Release. * Create Stories within Rational Team Concert (RTC) that contain the functional design. * Review, define, and evaluate the solution’s requirements, change requests, and new development requests. * Work with the Scrum Master to update, groom, and maintain the prioritized Product Backlog. * Develop, enhance, and manage acceptance criteria for product features. * Evaluate the complexity and scope of application improvement and enhancement requests. |
| **System Administrator (DEV, TEST, and SQA)** | * Install and maintain the hardware, system and application for the software development and test environments. * Configure user accounts for the appropriate testing environment. * Provide the build for the various environments. |
| **Developer** | * Construct and maintain the technology and development code within RTC. * Construct and maintain middleware technology. * Provide support during the different phases of testing. * Ensure that the solution is in compliance with the technical framework and architecture. * Test the development environment to verify proper configuration and operation. * Review and provide input to the Master Test Plan. * Execute unit testing and functional testing (as required), test results, and issue resolution. * Ensure that the solution meets the defined requirements. * Review and provide input to the Test Evaluation Summary and performance testing. |
| **Test Lead** | * Develop and maintain the Master Test Plan. * Develop and maintain the data elements within RQM. * Manage the formal test case design, formal system testing, and performance testing. * Identify and report testing related risks and recommend risk mitigation strategies. * Report and track defects/issues as they are discovered. * Perform retest on issues resolved in individual components, as required. * Provide input into the Test Evaluation Summary. * Enter data elements into RQM and update supporting test documentation. * Assist with the creation of test cases and test criteria for each applicable testing component. * Provide support to resources for System Integration and User Acceptance Testing (UAT) testing. * Participate in test related activities. * Perform sign-off of verification checklists and transition documents. * Ensure Test Tools are up-to-date and accurate with the latest increment/sprint information, if applicable. |
| **Test Analysts** | * Create and execute test cases and scripts within RQM. * Prepare test data as needed. * Perform end-to-end testing of all test cases executed to ensure complete system/integration testing. * Report all defects/issues in RTC as they are discovered. * Provide status of testing to Test Lead. * Continue to increase knowledge of business rules and requirements. * Provide input to the Master Test Plan. * Provide input to the Test Evaluation Summary Report. * Facilitate and provide guidance during UAT and/or User Functional Testing (UFT). |
| **Stakeholders / Users** | * Provide support for requirements and business workflow in which they may affect or be affected by the outcome. This includes pilot site testers and users. * Participate in UAT/UFT. * Validate/approve requirements. * Identify data sourcing for testing purposes. |

## Processes and References

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

* Agile Process Framework (Build Planning)
* VIP Methodology
* Test Planning
* Test Preparation
* Product Build
* Software Quality Assurance (SQA) Testing

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

* [ProPath](http://dns/process/home.aspx" \o "ProPath Home Page)
* [Section 508 Office Web Page](http://dns/index.asp)
* [Privacy Impact Assessment - Privacy Service](http://DNS/Privacy_Impact_Assessment.asp)
* [VA Release Readiness Office](file:///C:\Users\582061\Desktop\GenISIS\MTP\•%09https:\DNS\sites\OIT\epmo\TRS\SitePages\Forms\AllPages.aspx)
* [VIP Methodology](file:///C:\Users\582061\Desktop\GenISIS\MTP\•%09https:\DNS\sites\OIT\epmo\vip\Pages\HomePage.aspx)
* [ProPath Templates](http://DNS/process/propath)
* [Rational Tools Team Site](http://DNS/communities/OSCTM/toolsmgmt/Rational%20Tools/_layouts/viewlsts.aspx?BaseType=1)

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

* System Design Document (SDD): Version 6, Dated: 6/2017
* Requirements Traceability Matrix (RTM), Dated: 6/2017
* Risk Log: Version 2, Dated: 6/2017
* Genisis2 Build Plan: Version 1.1, Dated: 6/2017

# Items to Be Tested

This section identifies the functions and features that are with scope of the Genisis2 test effort.

## Overview of Test Inclusions

The components and features and combinations of components and features that will be tested in Genisis2 Releases 1-4 and have been identified and are listed below. Please note that names of the roles have changed for Release 2. Researcher is now Requestor and Data Manager is now called Data Destination Manager.

Role Based Testing

* + Requestor - Create, Modify, Track, Copy and Cancel Request
  + Data Destination Manager - Create, Modify, Track, Copy, Cancel, Approve Request
  + Data Source Manager – Accept or Deny Data Requests and Deliver Data Results
  + Genisis2 System Administrator – User Management, Create, Modify, Track, Copy, Cancel, Tracking and Management Reports (Release 4)
* E-mail Notification
* Backend Data Operations (Data-File Assessments, COPY Table function, Secure File Transfer)
* OI&T Compliance (Security, Design, Engineering & Architecture and Section 508)

Further details on the Genisis2 functionality and when they will be tested are in Tables 2-5. This document will be updated prior to each release to include more details on the features and functions that will be tested. Test cases will be written to thoroughly test each role and category for the new functionality that will be added within Genisis2. All data elements will also be added to RQM.

For information on project functionality, refer to the Genisis BRD, RSD, and SDD located on the [Genisis SharePoint Site](http://DNS/projects/Genisis/_layouts/viewlsts.aspx?BaseType=1) and within the Genisis Rational Instance.

## Overview of Test Exclusions

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

* Genisis 1.0 functions
* Network Capacity Testing
* Requirements that fall outside the Genisis2 project scope
* Regression testing of existing applications/utilities
* Honest Broker
* VINCI

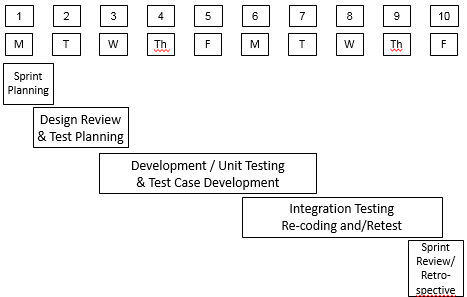
# Test Approach

The Agile Methodology is being followed by the Genisis2 Development and Test teams. Each release will be comprised of agile sprints, each having a specific focus and objective in driving toward the release. This section addresses the agile approach, compliance with VIP and how the Rational Tool Suite supports the full lifecycle effort for Genisis2. In addition, this section explains how the team will approach sprint planning, test planning and execution. RTC will be utilized by the Development Team for task management and both the Development and Test teams will utilize RTC for defect tracking.

Note: The test approach may be revised based on requirement clarifications and potential modifications to the scope as it may expand to include additional items in the backlog.

**Genisis2 Iterations – Agile Approach**

The Test Team will receive the build immediately after development is complete and unit tested. The application updates will be installed in the test environment for test execution of the new functionality. Depending on the build integrity, there may be interim builds to resolve configuration issues, defects, or if a build is deemed to be unacceptable for testing. Figure 1 illustrates our iteration approach:



**Figure 1: Genisis2 Iterative Agile Testing Approach**

**Rational Tools Suite**

The initial set of requirements contained in the RSD was decomposed into detailed requirements and user stories and entered into the Genisis2 Rational Requirements DOORS-NG or RM Repository. The user stories entered in RM were then manually created within RTC. Each user story created within RTC was linked to all of the associated requirements housed in RM which created the original link between RM and RTC. Each user story and associated requirement(s) were assigned to a Release Plan within RTC and each Release Plan was decomposed into sprints to support the delineation of tasks and work items.

The contents of the Release Plan, along with the SDD, serves as the basis for this Master Test Plan. Subsequently, this Master Test Plan is the basis for test planning, preparation, and execution data elements that are included within the Rational Quality Manager (RQM) Repository. RQM will be the repository of all test artifacts. Each test case within RQM will be linked to the applicable requirement within RM. This final link, in addition to the links between RM and RTC, demonstrate and support full traceability throughout the project lifecycle. This approach applies to all general system requirements, as well as the OI&T (DE&A, Section 508, Security, and Release Compliance) requirements.

**Genisis2 Releases 1-4**

In accordance with VIP, the Genisis2 Team will build, test, and deliver a release within 90 days. Most releases are comprised of 13 week cycles that includes four 2-week development sprints including testing, one 2-week stabilization sprint with SQA, one 2-week UAT sprint, and a 1-week release readiness sprint. During the release readiness sprint, and prior to VIP Critical Decision 2 (CD2), the Release Agent will review all required documentation and tasks to ensure and approve the readiness of the release. The review will also include the areas of OI&T Compliance (DE&A, Section 508, and Release Compliance). All test artifacts will be loaded and included within RQM and supported by this Master Test Plan.

The breakdown of features / functionality that will be tested and delivered in each Genisis2 release is identified in Tables 2-5. These tables will be updated at the start of each release based on any modifications that may occur during backlog grooming.

**Table 2: Release 1 Features and Functionality**

| **Role / Work stream** | **Features / Functionality** |
| --- | --- |
| **Researcher (UI)** | * Create Data Request * Modify Data Request * Track Data Request * E-mail notification for status changes * E-mail notification Genisis2 Landing zone (manual) |
| **Data Manager (UI)** | * Create Data Request * Modify Data Request * Approve Request * Track Data Request * E-mail notification for status changes (automatic) * E-mail notification VINCI Landing zone (manual) |
| **Back end** | * Data Operations Framework * Basic Data File Assessment (counts for number of rows and missing data) |
| **Security** | * Basic login (no PIV) * Role based access |

**Table 3: Release 2 Features and Functionality**

| **Role / Work stream** | **Features / Functionality** |
| --- | --- |
| **Requestor (UI)** | * Regression Create/Modify/Track Data Request * Automatic e-mail notification for status changes * Add Comments |
| **Data Destination Manager (UI)** | * Regression Create, Modify, Track, Approve, Deny, Return Data Request * Automatic e-mail notification for status changes * Add Comments |
| **History (UI)** | * Capture high level and detailed history information for each request. |
| **Comments (UI)** | * Provide the ability to add comments throughout the lifecycle of a data request |
| **Back end** | * Secure transfer of data from VINCI * Data File Assessment (file checking) * COPY Table Function (VINCI to Genisis) |
| **Security** | * Integration with VA’s Active Directory * Role-based access with two-factor authentication |

**Table 4: Release 3 Features and Functionality**

| **Role / Work Stream** | **Features / Functionality** |
| --- | --- |
| **Requestor** **(UI)** | * Regression Create, Modify, Track, Copy, Cancel Data Request * Automatic e-mail notification * VINCI documentation view |
| **Data Destination Manager (UI)** | * Regression Create, Modify, Track, Copy, Cancel, Approve Data Request * Automatic e-mail notification * Data Management utility library |
| **Data Source Manager (UI)** | * Accept Data Request * Deny Data Request (Request Cannot be Fulfilled) * Deliver Results |
| **History (UI)** | * Capture high level and detailed history information for each request. |
| **Comments (UI)** | * Provide the ability to add comments throughout the lifecycle of a data request |
| **Back end** | * Regression basic data file assessment for number of rows and missing data * Automatic detection (notification) of data from VINCI |
| **Security** | * Regression Role-based access with Two-Factor Authentication |

**Table 5: Proposed Release 4 Features and Functionality**

| **Role / Work Stream** | **Features / Functionality** |
| --- | --- |
| **Requestor (UI)** | * Regression Create, Modify, Track, Copy, Cancel Data Request, Automatic e-mail notification * VINCI documentation view * Usability Compliance |
| **Data Destination Manager (UI)** | * Regression Create, Modify, Track, Approve, Copy, Cancel Data Request, Automatic e-mail notification. Data Management utility library * Usability Compliance |
| **Genisis2 System Administrator (UI)** | * Regression Test Tracking Reports * User management * Usability Compliance * Additional Reports |
| **Back end** | * Regression basic data file assessment for number of rows and missing data) * Automatic detection (notification) of data from VINCI * Secure transfer of data from VINCI * Enhanced Data Operations (file checking) * Enhanced Performance |
| **Security** | * Regression Role-based access with Two-Factor Authentication * More Enhanced Security |

## Product Component Test

Prior to delivery of the sprint build to the testers, the developers will perform Product Component Testing (also known as Unit Testing) on code they have developed. These activities will be performed in the DEV environment. Once the stability of the new functionality that supports the user story and associated requirements is confirmed, the results will be communicated to the Scrum Master and Development Project Manager. Once approvals to move forward have been given, the results and source code will be uploaded into RTC and a new build will be readied for the Test Team.

## Component Integration Test

After the Development Team has completed Product Component Testing for a sprint, the Development Team will deliver the build to the Test Team to perform Component Integration Testing. This will take the initial form of Smoke Testing to confirm the stability and viability of the delivered build before proceeding to test execution of the delivered component. This testing will be done within the TEST environment. Test results will be documented within RQM and the testers will make an assessment based on those test results whether to continue with system testing or return the build to development for additional work.

## System Tests

As new builds become available and stable enough to promote to the AWS-Test environment, System Testing will be executed by the Test Team in the AWS-Test environment. High- level test scripts will be created based on available functionality delivered from the Development Team for each component and sprint. Testing will take the form of exercising the high-level scripts when appropriate, and supported by “ad hoc” testing due to the frequent development cycles and potentially rapidly responsive design changes. Test results will be available after each component has been tested within RQM and via the Genisis2 Release Excel Spreadsheet.

## User Acceptance Testing

UAT will be performed upon completion of System Testing and SQA. Testing will be performed for the release by stakeholders / users within the VA-SQA environment. The Test Team will provide access to the test cases within RQM if requested, but the user’s scenarios should be developed by the users. The users will provide results for reporting progress and completion of UAT testing using a method that will be established for each release. The Test Team will also provide support to assist in documentation of the test results, if required, and will capture this information with RQM or a spreadsheet uploaded to Rational for test results and RTC for any defects identified.

## Enterprise System Engineering Testing

Enterprise System Engineering (ESE) Testing is typically conducted by the ESE Testing Services Team and supported by the Genisis2 Development Team. The Genisis2 Test Lead will schedule a kick off meeting with ESE Testing Services to plan and coordinate any needed support for release readiness in accordance with the VIP Methodology. This will include the provision of documentation, access to environments, and any additional tasks as required to support their Statistic and Dynamic Testing requirements. This section of the Master Test Plan will be updated to include any issues and/or risks identified by ESE once they have completed their review and/or testing for each release.

## Performance Testing

This section will be updated in a future release to identify the approach to performance testing based on system requirements. The update will include details about tools, if applicable, team POCs, any special requirements and dependencies.

# 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
* Identification and documentation of software defects (e.g., RTC)

Dynamic analysis deals with specific methods for ascertaining software quality through actual executions (e.g., 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
* Performance Testing
* User Acceptance Testing

## Risk-based Testing

The Genisis2 Test Team will develop and execute test cases and/or scripts in accordance with the following functional priorities:

* Specific testing recommended by the Genisis2 Development Team or testing that requires additional/alternative testing resources.
* Specific functionality necessitated by issues that the Genisis2 Development Team encounters.
* Specific testing requirements recommended by the developers, SQA Analysts and / or users which were not initially identified will result in an update to the User Story documentation to ensure there is traceability from the user stories to the test cases.
* Specific recommendations documented within the Enterprise Systems Engineering Risk Analysis and Testing Scope Report.

## Enterprise Testing

The Genisis2 Project Team will work with the Design & Architecture Compliance Group (DE&A) to identify the OI&T DE&A Epics required to ensure Genisis2 is in compliance with VA’s Architecture standards. The applicable DE&A User Stories will be captured and moved into the Genisis Rational Instance. Resources from the DE&A will work with members of the Genisis2 Development, Test and Configuration Management Team to provide information in the form of the Genisis2 System Design Document (SDD) that will validate the OI&T DE&A Compliance requirements and user stories. This will ensure the system meets VA Enterprise standards and support the required traceability from RM through RQM for Release Readiness.

### Security Testing

The Genisis2 Team contacted a member of the Cyber Security Policy and Compliance Group (CSPC) to discuss the OI&T Security Epics. CSPC is in the process of reviewing and rewriting the OI&T Security Epics, Sub-Epics and User Stories so we were advised to wait until early January 2017. The Genisis2 Team will meet at that time to review and identify the Security User Stories applicable to Genisis2. This information will be used to develop test cases that validate the system and OI&T security requirements and ensure Release Readiness.

### Privacy Testing

The Genisis2 Team contacted a member of the CSPC to discuss the OI&T Security Epics related to the Privacy and Security Rule provisions of the Health Insurance Portability and Accountability Act (HIPAA). This information will be used to support tests to ensure that (1) veteran and employee data are adequately protected, and (2) systems and applications comply with the Privacy and Security Rule provisions of HIPAA to secure an Authority to Operate (ATO). Genisis2 is currently covered under the Region 4 ATO; tests will be limited to WASA and Fortify scans executed by the Development Team.

### Section 508 Compliance Testing

The Development and Test Teams are responsible for ensuring that Genisis2 functionality is usable from the keyboard, while the Section 508 Program Office is responsible for performing independent compliance testing with assistive technology.

The project must obtain sign-off from the Section 508 Program Office that compliance testing was performed. For more information, contact the Section 508 Program Office at [DNS](mailto:DNS).

The Genisis2 Project Team will meet with representatives of the Section 508 Compliance office to determine how Genisis2 will be certified and identify the approved tools for use. The Genisis2 Team will perform internal 508 testing to determine readiness for the 508 audit. Once completed, the Genisis2 team will submit a request to the VA-508 team to perform independent testing and provide 508 approval.

The following is a list of tests which need to be performed to be in compliance with Section 508:

* Can you use the keyboard instead of the mouse?
* Does the cursor move in a logical order or flow?
* Do the elements do what they are supposed to do?
* Is there alternate text for all non-text elements?
* Does the link text explain what the link does?
* Are there captions for audio and visual elements or transcripts for audio only?
* Is color the only means of identification of elements on a page?
* Are documents organized so they are readable without requiring an associated style?
* Are there server side image maps or client side image maps?
* Are tables coded properly?
* Does your website have frames?
* Does the screen flicker with a frequency greater than 2 Hz and lower than 55 Hz?
* Are there text-only pages for information that cannot be made compliant in any other way?
* Is the script language in a readable fashion for assistive technology users?
* Is there a link for software downloads?
* Are there electronic forms?
* Is there a way for the user to skip navigation functions/sidebar and go straight to the content?
* If a timed response is used, is the user prompted to request more time?

### Multi-Divisional Testing

Since Genisis2 will be deployed within PITT at one site, this project does not need to address any specific multi-divisional requirements; therefore, multi-divisional testing is not required.

## Performance and Capacity Testing

The Genisis2 Development Team will perform quantitative tests that will measure the response time at which a system functions to ensure that the application meets the specifications defined within the non-functional requirements.

## Test Types

Table 6 lists the types of tests to be performed on the Genisis2 application, as appropriate.

**Table 6: Test Types**

| Test Types | Party Responsible |
| --- | --- |
| **Access control testing** | Development / Test Teams |
| **Build verification testing** | Development Team |
| **Compliance testing** | Test Team |
| **Component integration testing** | Development Team |
| **Configuration testing** | Development Team |
| **Data and database integrity testing** | Development / Test Teams |
| **Documentation testing** | Development / Test Teams |
| **Error analysis testing** | Development / Test Teams |
| **Exploratory testing** | Test Team |
| **Failover testing** | N/A |
| **Installation testing** | Development / Test Teams |
| **Integration testing** | Development / Test Teams |
| **Migration testing** | Development Team |
| **Multi-divisional testing** | N/A |
| **Parallel testing** | Development / Test Teams |
| **Performance monitoring testing** | Test Teams |
| **Performance testing** | Development Teams |
| **Performance - Benchmark testing** | Development Teams |
| **Performance - Contention testing** | Development Team |
| **Performance - Endurance testing** | Development Team |
| **Performance - Load testing** | Development Team |
| **Performance - Profiling testing** | Development Team |
| **Performance - Spike testing** | Development Team |
| **Performance - Stress testing** | Development Team |
| **Privacy testing** | Development / Test Teams |
| **Product component testing** | Development Team |
| **Recovery testing** | Development / Test Teams |
| **Regression test** | Test Team |
| **Risk based testing** | Test Team |
| **Section 508 compliance testing** | Test Team / 508 Compliance Group |
| **Security testing** | Test Team |
| **Smoke testing** | Development / Test Teams |
| **System testing** | Test Team |
| **Usability testing** | Test Team |
| **User Functionality Testing** | Test / Users Teams |
| **User interface testing** | Development / Test / User Teams |

## Productivity and Support Tools

The Genisis2 Test Team will utilize the tools listed in Table 7 to support test management, test execution, defect tracking, and test reporting activities.

**Table 7: Tool Category or Types**

| Tool Category or Type | Tool Brand Name | Vendor or In-house | Version |
| --- | --- | --- | --- |
| Test Management | RQM | In-house | Version 6 |
| Defect Tracking | RTC | In-house | Version 6 |
| Test Coverage Monitor or Profiler | Rational DOORS-NG & RQM | In-house | Version 6 |
| Project Management | Microsoft Project | In-house | 2013 |
| Performance Testing | TBD | TBD | TBD |
| Configuration Management | RTC | In-house | Version 6 |
| Functional Test Automation | Rational Functional Tester | In-house | TBD |

# Test Criteria

## Process Reviews

The Master Test Plan undergoes two reviews:

* Peer Review – upon completion of the Master Test Plan and is updated by both the Development and Test Teams.
* Formal Review – Presentation to the VA PM after the Development and Test Lead approves the Master Test Plan.

The Master Test Plan serves as input for data elements entered into RQM. Both the Master Test Plan and the data elements contained in RQM support the VIP processes for traceability of requirements throughout the project’s lifecycle and data required to support decisions required for the Critical Decision 2 (CD2) and Release Management.

For more background information on the reviews associated with testing, see the Product Build, Test Preparation, and Independent Test and Evaluation processes. Even though ProPath Processes are no longer required, they are still used for best practice guidelines.

## Pass/Fail Criteria

Incidents identified during the execution of this test plan will be evaluated to determine their severity. All incidents will be recorded in RTC:

* + 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.

1) 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.

2) 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 prior to release.

## 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 VA Network Servers or CAG.

# Test Deliverables

The Test Deliverables listed in Table 8 lists the test deliverables for the Genisis2 project.

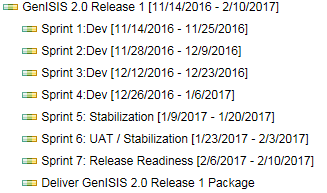
**Table 8: Test Deliverables**

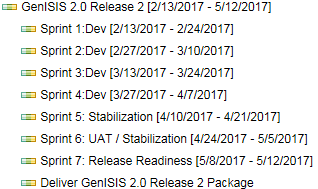
|  |  |
| --- | --- |
| **Test Deliverables** | **Responsible Party** |
| **Master Test Plan (MTP)** | Test Lead |
| **Master Test Plan (Rational RQM)** | Test Lead |
| **Master Test Plan Checklist** | Test Lead |
| **Iteration Test Plans (Rational RQM)** | Test Lead & Test Team |
| **Test Schedule (included within (RTC)** | Test Lead |
| **Test Cases/Test Scripts (Rational RQM)** | Test Team |
| **Test Data** | TBD |
| **Test Environment** | Genisis 2 System Administrator & Test Team |
| **Integrated Test Environment** | Genisis2 System Administrator, Development & Test |
| **Traceability Matrix**  **(Rational DOORS–NG & RQM)** | Test Team |
| **Test Defect Logs (Rational RTC)** | Test Team |
| **Test Execution Logs (Rational RQM)** | Test Team |
| **Test Evaluation Summaries** | Test Lead & Test Team |

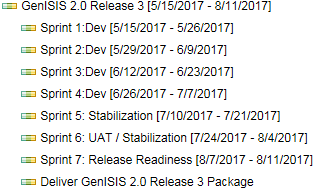
# Test Schedule

The overall project schedule is being managed within the Genisis2 Rational Instance; and specifically, within RTC and RQM.

The following is a snapshot in the Genisis Rational Instance and includes the projected schedule for Releases 1-4.









**Genisis2 will not be released nationally. The Release Site for Genisis2 is PITC.**

Table 9 lists the milestones for testing Genisis2.

**Table 9: Testing Milestones**

| **Testing Milestones** | **Responsible Party** |
| --- | --- |
| **Complete Master Test Plan** | Test Lead / Test Engineer |
| **Complete Requirements Traceability Matrix** | Requirements Manager &  Test Lead / Engineer |
| **Unit Testing** | Development Team |
| **Functional Testing** | Test Team |
| **DE&A Testing** | Development & Test Teams |
| **Security Testing** | Development & Test Teams |
| **Performance Testing** | Development Team |
| **Section 508 Testing** | Test Team |
| **User Acceptance Testing** | Users with Development & Test Support |
| **Test Evaluation Summary** | Test Team |

# Test Environments

The Configuration Manager (CM) is responsible for managing the releases for all Genisis2 project test environments. The DBA is responsible for controlling and maintaining all Genisis2 project test environments. Unplanned changes to the test environments may introduce new test incidents, alter the expected test results, and thus, invalidate the results of the Test Cases. The CM is responsible for builds, national patch installs, set-up, and configuration change support. The Internet Explorer (IE) Team is responsible for configuration and maintenance of the interface engines.

Test Environment Quality Gate: The test system is stood up and source code is integrated into a change-controlled environment by following the Configuration Management Plan procedures prior to testing.

## Test Environment Configurations

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 invalidate the test cases. Successful testing requires controlled access to the test environment, an environment that replicates the field environment as closely as possible.

Development testing will be conducted in the development environment. System and integration test environments should simulate the production environment where the software will be executed. One dedicated test environment will be set up and ready the test team. This test environment will also be used by the SQA Team for system and integration testing. All promotions to the test environment will be managed through a change management process and access to this environment will be managed by the Genisis2 Development and Test Team.

A Genisis2 test environment will be utilized in order to perform integration testing for modules and components they are ready to be considered for promotion and release. The Test Team will maintain this account.

The Test Environment Configurations need to be provided and supported for the Genisis2 project. Table 10 provides the test environment information.

**Table 10: Necessary Genisis2 Test Environment**

|  |  |
| --- | --- |
| **Configurations Configuration Name** | **Description** |
| Server Name: DNS  Database Names: jbpmdb (JBPM), Genisisdb (app database)  Server Location: Austin, TX | Database Tier: MS SQL Server 2012 |

## AWS Test Environment Configuration

The Test Team will work with development and the system administrators to set up users and data required for the SQA environment. Specifics are identified in Sections 8.3 and 8.4. The test system will not contain “live” patient data. Table 11 provides information regarding the temporary test environment.

**Table 11: Amazon Web Services (AWS) Genisis2 Temporary Test Environment**

| **Resource** | **Quantity** | **Name and Type** |
| --- | --- | --- |
| **Database Server** | 1 | IP |
| **Network or Subnet** | 1 | IP |
| **Database Name** | 2 | jbpmdb (JBPM), Genisisdb (app database) |
| **Test Repository** | 1 | IP |
| **Test and Development GFEs** | 8 | (6) i7 processor (Development) (2) i5 processor (Test) |

## VA (AITC) Test Environment Configurations

Functionally, exact instances of the Genisis2 environment will be created and maintained. The Test Team will use one instance for unit and integration testing, while one instance will be reserved for system testing by the SQA Team.

The test system will not contain “live” patient data.

The specific elements of the test system may not be fully understood in early iterations, so this section may be completed/updated over time. The test system should simulate the production environment as closely as possible, scaling down concurrent access and database size, and so forth, if and where appropriate. Table 12 sets forth the system resources for the test effort presented in this Master Test Plan, and will be tailored as needed.

Table 12: System Hardware Resources

| Resource | Quantity | Name and Type |
| --- | --- | --- |
| Database Server | 1 | DNS |
| Network or Subnet | TBD | IP |
| Server Name | 1 | DNS |
| Database Name | 1 | IP |
| Test Repository | 1 | DNS |
| Address | 1 | IP |
| Test Development PCs | TBD | Using a combination of GFEs and CAG Access to VA |

## Base Software Elements in the Test Environments

Table 13 describes the base software elements that are required in the test environment for this Master Test Plan. Software elements will be adjusted, as appropriate. If necessary, software patches will be provided or referenced in the table.

Table 13: Software Elements

| Software Element Name | Version | Type and Other Notes |
| --- | --- | --- |
| **Linux** | RHEL | Operating System |
| **Windows Server** | 2008 R2 | Operating System |
| **Internet Explorer 11** | Version 11 | Internet Browser |
| **MS Outlook 2010** | Version 14 | Email Client software |

# Staffing and Training Needs

Table 14 indicates the number of personnel resources needed to plan, prepare, and execute the testing tasks.

Table 14: Staffing Resources

| Testing Task | Quantity of Personnel Needed | Test Process | Duration (Days) |
| --- | --- | --- | --- |
| **Develop the Master Test Plan** | 1 FTE | Test Planning | 5 days |
| **Set up Genisis in the RQM Instance** | 1 FTE | Test Planning | 2 days |
| **Create Test Plans for each Release within Rational** | 1.0 FTE | Test Planning | 2 days |
| **Establish the TEST Environment** | 1.5 FTE | Test Preparation | 2 days |
| **Create test cases and/or scripts for each Sprint** | 3 FTE | Test Preparation | 5 days |
| **Enter data elements (test cases/scripts) into RQM, create test records and link scenarios to requirements in RM for traceability (RTM). Perform after final development sprint.** | 1 FTE | Test Preparation & Test Execution | 10 days |
| **Execute tests for each module per sprint** | 3 FTE | Test Execution | 4 days |
| **Document Test Results** | 3 FTE | Test Reporting | 1 day |

Table 15 identifies the training needs required to execute the activities outlined in the Master Test Plan.

Table 15: Training Needs

| **Name** | **Training Need** | **Training Option** | **Estimated Training Hours** |
| --- | --- | --- | --- |
| **Stakeholders**  **(PM, COR)** | IBM Rational (RM, RQM) | VA-TMS training | 8 hours |
| **Architect & Developers** | IBM Rational (RTC) | VA-TMS training | 8 hours |
| **Scrum Master** | IBM Rational (RTC) | VA-TMS training | 8 hours |
| **Business Analyst** | IBM Rational (RM) | VA-TMS training | 8 hours |
| **Subject Matter Experts (SME)** | IBM Rational (RM, RQM) | VA-TMS training | 8 hours |
| **Test Engineers** | IBM Rational (RQM) | VA-TMS training | 8 hours |
| **UAT/UFT Testers** | IBM Rational (RQM) | VA-TMS training | 8 hours |
| **Newly on-boarded**  **Testers/Developers** | IBM Rational (RTC & RQM) | VA-TMS training &  Hands-on | 8 hours |
| **Functional Testers (2)** | IBM Rational Functional Tester | VA Sponsored Training | 16 hours |

# Risks and Constraints

Risks associated with testing are potential problems/events that may cause damage to the software, system, operating systems, schedule, scope, budget, or resources. The risk log was taken into consideration in the development of this test plan. The risks outlined may impact the scope and the schedule, necessitating a deviation from this test plan. Table 16 lists the risks identified for this test plan. The risks identified in this Master Test Plan can be found in the risk log and is recorded and tracked in RTC.

Table 16: Risk List

| **Risk Description** | **Potential Impact** | **Mitigation / Avoidance** |
| --- | --- | --- |
| **Requirement specification updates late in the development cycle** | High | * Establish and enforce requirement completion dates. * Provide effective change management. * Update test plan and test cases as requirements are known. * Encourage the project team to address high-risk high-need requirements early. |
| **Lack of availability of UAT personnel causes schedule delays** | Medium | * Communicate with the users about project schedule. * Work with users to ensure adequate support. * Identify backup support staff, if applicable |
| **Lack of availability of SQA personnel causes schedule delays** | Medium | * Communicate with the SQA personnel about project schedule. * Work with SQA to ensure adequate support. * Identify backup support staff, if applicable |
| **Critical defect discovered during SQA or UAT requiring remediation** | High | * Test software and devices thoroughly prior to SQA and UAT. * Create reports and provide escalation when issues are not addressed in a timely manner. |
| **Lack of communication between the teams** | High | * Identify key personnel on each team that can coordinate communication and efforts between the teams (e.g., Genisis, VINCI). |

# 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 Test Analyst will capture some of the measures specified above.

Attachment A: Approval Signatures

REVIEW DATE: June 16, 2017

Signed: Date:

Katie A, Thomas – Project Manager

Signed: Date:

Saiju Pyarajan – Business Sponsor

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 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. |
| 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 number 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 Health*e*Vet (H*e*V)-VistA operating procedures and loads the latest .jar version onto a live copy of VistA and H*e*V-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 – Profiling 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 – 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 HIPAA. |
| Product Component Testing | Product Component Testing (also known as 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 | 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. |

Template Revision History

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| November 2015 | 1.18 | Expanded Section 4.3 to better describe responsibilities for 508 compliance. | Channing Jonker |
| October 2015 | 1.17 | Corrected broken link to 508 URL. | Channing Jonker |
| 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 |