<Enter Project Name Here>

# Master Test Plan



<Month><Year>

**Version** <#.#>

Department of Veterans Affairs

This template contains a paragraph style called Instructional Text. Text using this paragraph style is designed to assist the reader in completing the document. Text in paragraphs added after this help text is automatically set to the appropriate body text level. For best results and to maintain formatting consistency, use the provided paragraph styles. Delete all instructional text before publishing or distributing the document Revision History.

This template conforms to the latest Section 508 guidelines. The user of the template is responsible to maintain Section 508 conformance for any artifact created from this template.

Revision History

| Date | Version | Description | Author |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

Place latest revisions at top of table.

The Revision History pertains only to changes in the content of the document or any updates made after distribution. It does not apply to the formatting of the template.

Remove blank rows.

Table of Contents

[Introduction 1](#_Toc163493381)

[Purpose 1](#_Toc163493382)

[Test Objectives 1](#_Toc163493383)

[Roles and Responsibilities 2](#_Toc163493384)

[Processes and References 2](#_Toc163493385)

[Items To Be Tested 3](#_Toc163493386)

[Overview of Test Inclusions 3](#_Toc163493387)

[Overview of Test Exclusions 3](#_Toc163493388)

[Test Approach 3](#_Toc163493389)

[Product Component Test 3](#_Toc163493390)

[Component Integration Test 3](#_Toc163493391)

[System Tests 3](#_Toc163493392)

[User Functionality Test 4](#_Toc163493393)

[Enterprise System Engineering Testing 4](#_Toc163493394)

[Initial Operating Capability Evaluation 4](#_Toc163493395)

[Testing Techniques 4](#_Toc163493396)

[Risk-based Testing 4](#_Toc163493397)

[Enterprise Testing 4](#_Toc163493398)

[Security Testing 4](#_Toc163493399)

[Privacy Testing 5](#_Toc163493400)

[Section 508 Compliance Testing 5](#_Toc163493401)

[Multi-Divisional Testing 5](#_Toc163493402)

[Performance and Capacity Testing 5](#_Toc163493403)

[Test Types 6](#_Toc163493404)

[Productivity and Support Tools 7](#_Toc163493405)

[Test Criteria 7](#_Toc163493406)

[Process Reviews 7](#_Toc163493407)

[Pass/Fail Criteria 8](#_Toc163493408)

[Suspension and Resumption Criteria 8](#_Toc163493409)

[Test Deliverables 8](#_Toc163493410)

[Test Schedule 9](#_Toc163493411)

[Test Environments 9](#_Toc163493412)

[Test Environment Configurations 9](#_Toc163493413)

[Base System Hardware 9](#_Toc163493414)

[Base Software Elements in the Test Environments 10](#_Toc163493415)

[Staffing and Training Needs 11](#_Toc163493416)

[Risks and Constraints 11](#_Toc163493417)

[Test Metrics 12](#_Toc163493418)

[Attachment A – Approval Signatures 13](#_Toc163493419)

[Appendix A - Test Type Definitions 14](#_Toc163493420)

## Introduction

### Purpose

Briefly state the purpose of the testing initiative.

Include that this test plan will:

* Document the overall testing process.
* Describe the Test Strategy including defining the test levels and types of tests planned.
* Include testing activities to be performed.
* Document who will perform the test activities.

For projects which include Vista Patches:

The patch identifier for this Test plan is <Patch Number e.g. FB\*3.5\*124>.

For non-Vista projects:

The version of the software product, e.g. VA-ONCE: P040, WEAMS: 3\_0\_17, Chapter 33: 6.4.

### Test Objectives

Tailor the test objectives as appropriate.

This Master Test Plan supports the following objectives:

* To provide test coverage for 100% of the documented requirements
* To provide coverage for System/ Software Design Document elements
* To execute 100% of the test cases during User Functionality Testing
* To execute 100% of the Performance testing
* To create, maintain and control the test environment
* Add other objectives as needed

### Roles and Responsibilities

Customize the table below according to the roles that support the execution of the Master Test Plan.

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

Table : Roles and Descriptions

| Role | Description |
| --- | --- |
| Development Team  | Persons that build or construct the product/product component. |
| Development Manager | Person responsible for assisting with the creation and implementation of the Master Test Plan. |
| Program Manager | Person that has overall responsibility for the successful planning and execution of a project; person responsible for creating the Master Test Plan in collaboration with the Development Manager.  |
| Stakeholders | Persons that hold a stake in a situation in which they may affect or be affected by the outcome. |
| Test Analyst | Person responsible for ensuring full execution of the test process to include the verification of technical requirements and the validation of business requirements.  |
| Test Lead | An experienced Test Analyst or member of the Test Team that leads and coordinates activities related to all aspects of testing based on an approved Master Test Plan and schedule. |
| Test Team/Testers | Persons that execute tests and ensure the test environment will adequately support planned test activities. |
| Test Environment Team | Persons that establish, maintain, and control test environments. |

Remove blank rows.

### Processes and References

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

* Test Preparation
* Product Build
* Independent Test and Evaluation

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

* Process Asset Library (PAL)
* Section 508 Office Web Page
* Privacy Impact Assessment - Privacy Service

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

* Business Requirement Document (BRD) Version <#.#>, Date <Month, Year>
* Requirements Specification Document (RSD) Version <#.#>, Date <Month, Year>
* System Design Document (SDD) Version <#.#>, Date <Month, Year>
* Requirements Traceability Matrix (RTM) Version <#.#>, Date <Month, Year>
* Risk Log Version <#.#>, Date <Month, Year>

## Items To Be Tested

List those test items - software, hardware, and supporting product elements  that serve as targets for testing. A test item may include source code, object code, job control code, control data, documentation, or a collection of these.

### Overview of Test Inclusions

Provide a high-level list of the major target test items. This list should include both items produced directly by the project Development Team and, if applicable, as well as vendor-supplied products being integrated into the information system or application. Refer to the Requirements Specification Document (RSD) to identify the requirements needed for testing items that those products rely on; For example, basic processor hardware, peripheral devices, operating systems, third-party products or components, and so forth. Consider grouping the list by category and assigning relative importance to each motivator.

The following components and features and combinations of components and features will be tested:

### Overview of Test Exclusions

Identify any items specifically excluded from testing.

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

## Test Approach

The Test Approach is the implementation of the Test Strategy. 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 the PAL.

### Product Component Test

Briefly describe how the Developers perform Product Component Test, also known as Unit Test. Identify the responsible roles. For more information, see the Product Build process in the PAL.

### Component Integration Test

Briefly describe how the Developers perform Component Integration Test. Identify the responsible roles. For more information, see the Product Build process in the PAL.

### System Tests

Briefly describe how the system or application will be tested during System Tests. At a high level specify any testing requirements, such as, test environment, hardware, test data, or dependencies. For more information, see the Product Build process in the PAL.

### User Functionality Test

Briefly describe how the system or application will be tested during User Functionality Test. At a high level specify any testing requirements, such as, point of contact, test environment, test data, hardware, or dependencies. For more information, see the Product Build process in the PAL.

### Enterprise System Engineering Testing

Specify how the Development Team will support Enterprise System Engineering (ESE) testing, the development team point of contact, and any special testing requirements and dependencies, including Performance Testing. Include intended testing process, plans for test scripts, and likely test scenarios. For more information on ESE testing, see the ESE Website. For more information on ETS performance testing, see the Independent Test and Evaluation process in the PAL.

### Initial Operating Capability Evaluation

Briefly describe how the Development Team will support the Test Sites during Initial Operating Capability Evaluation. Initial Operating Capability Evaluation was formerly known as Field Testing. For more information, see the Release Management process in the PAL.

## Testing Techniques

Testing Techniques describes the approach to risk-based testing, requirements for enterprise testing, test types, iterations, and tools that are used to test the designated test items as applicable.

### Risk-based Testing

Describe the potential risks that may cause the system to not meet reasonable user and customer expectations of quality. Risk-based testing is a technique for prioritizing testing based on testing the highest risk items first and continuing to test down the risk prioritization ladder as the testing schedule permits. Describe how the identified risks have been covered in the testing effort. For example, a table may be created to identify which test type or which test cases will be executed to address the identified risks.

### Enterprise Testing

Cite how the project testing covers the enterprise requirements. Enterprise requirements include security, privacy, Section 508 Compliance requirements, and Multi-divisional requirements.

#### Security Testing

Develop tests to validate the security requirements and to ensure readiness for the independent testing performed by the Security Assessment Team as used by the Independent Test and Evaluation Process. This test type validates the requirements specified in “Security Specifications” in the Requirements Specification Document found in the Requirements process in the PAL.

For more information on security testing, contact the Facility Information Security Officer.

#### Privacy Testing

Develop 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 the Health Insurance Portability and Accountability Act (HIPAA). The Privacy Impact Assessment (PIA) is a required component for the Assessment and Authorization (security) Package.

This test type validates the requirements specified in “Usability Specifications” in the Requirements Specification Document found in the Requirements process in the PAL. For more information, see the Privacy Service Home Page.

#### Section 508 Compliance Testing

Section 508 Compliance Testing is required for all applications.

Development Team is responsible for ensuring that product functionality is accessible and works with adaptive technology. Section 508 Program Office provides consultation on how to implement and test Section 508 compliant solutions, tools to conduct the testing, and training on how to use the tools and other aspects of Section 508. This test type validates the requirements specified in “Usability Specifications” in the Requirements Specification Document found in the Requirements process in the PAL.

The project must submit proof of compliance to Section 508 Office. For more information, contact the Section 508 Program Office.

#### Multi-Divisional Testing

Multi-Divisional Testing is required to ensure 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 The Requirements Specification Document defines the multi-divisional requirements for each application or system.

The Development Team is responsible for verifying and validating that the application or system complies with the multi-divisional requirements. This test type validates the requirements specified in “Multi-Divisional Specifications” in the Requirements Specification Document found in the Requirements process in the PAL.

### Performance and Capacity Testing

Develop tests to ensure the application will perform as expected under anticipated user loads, and typical business transactions respond in a timely manner. During the test execution, the System Under Test (SUT) is actively monitored for any issues that could affect application performance, and to verify the hardware environment is adequately sized.

This type of testing covers the requirements specified in the “Performance Specifications” in the Requirements Specification Document found in the Requirements process in the PAL.

### Test Types

Test types are a group of test activities aimed at testing a component or system regarding one or more interrelated quality attributes. A test type is focused on a specific test objective, i.e., reliability test, usability test, regression test etc., and may take place on one or more test levels or test phases. Specify the Test Types to be performed and the party responsible for performing the test. Delete from the table any test type that does not apply.

Table : Test Types

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

Remove blank rows.

### Productivity and Support Tools

Add or delete tools as appropriate.

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

Table : Tool Category or Types

| Tool Category or Type | Tool Brand Name | Vendor or In-house | Version |
| --- | --- | --- | --- |
| Test Management |  |  |  |
| Defect Tracking |  |  |  |
| Test Coverage Monitor or Profiler |  |  |  |
| Project Management |  |  |  |
| Performance Testing |  |  |  |
| Configuration Management |  |  |  |
| DBMS tools |  |  |  |
| Functional Test Automation |  |  |  |
| Other |  |  |  |

Remove blank rows.

### Test Criteria

#### Process Reviews

The Master Test Plan undergoes two reviews:

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

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

#### Pass/Fail Criteria

Pass/Fail criteria are decision rules used to determine whether a test item (function) or feature has passed or failed a test.

Specify the criteria to be used to determine whether the test items have passed or failed testing.

#### Suspension and Resumption Criteria

Suspension Criteria are the criteria used to (temporarily) stop all or a portion of the testing activities on the test items. Resumption Criteria are the testing activities that must be repeated when testing is re-started after a suspension.

Specify the suspension and resumption criteria that will guide test execution.

## Test Deliverables

The Test Deliverables listed below represent some possible deliverables for a testing project. The Test Deliverables table may be tailored to meet project needs. Do not include Delete any listed test deliverable that is not used by the Product Build, Test Management, and Independent Test and Evaluation processes.

Table 4 lists the test deliverables for the {project name here} project.

Table : Test Deliverables

| Test Deliverables | Responsible Party |
| --- | --- |
| Master Test Plan | {Name}, Role |
| Performance Test Plan | {Name}, Role |
| Iteration Test Plans (when appropriate) | {Name}, Role |
| Test Execution Risks  | {Name}, Role |
| Test Schedule | {Name}, Role |
| Test Cases/Test Scripts | {Name}, Role |
| Test Data | {Name}, Role |
| Test Environment | {Name}, Role |
| Test Evaluation (including performance test results) | {Name}, Role |
| Traceability Report or Matrix | {Name}, Role |

Remove blank rows.

## Test Schedule

List the major testing milestones. When appropriate, reference other workflow documentation or tools, such as the Project Management Plan, or Work Breakdown Structure (WBS.) Put a minimum amount of process and planning information within the Master Test Plan in order to facilitate ongoing maintenance of the test schedule.

Table : Testing Milestones

| Testing Milestones | Responsible Party |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Remove blank rows.

## Test Environments

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

### Test Environment Configurations

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

In order to ensure the verification and validation of applications and systems requiring multi-divisional capabilities, be sure to configure the test environments as multi-divisional environments. For more information, see section 3.3.4 in this document.

The party or parties responsible for configuring and maintaining the test environments are: {person responsible and group}.

### Base System Hardware

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

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

Table : System Hardware Resources

| Resource | Quantity | Name and Type |
| --- | --- | --- |
| Database Server |  |  |
| Network or Subnet |  | TBD |
| Server Name |  | TBD |
| Database Name |  | TBD |
| Client Test PCs |  |  |
| Include special configuration requirements |  | TBD |
| Test Repository |  |  |
| Network or Subnet |  | TBD |
| Server Name |  | TBD |
| Test Development PCs |  | TBD |

Remove blank rows.

### Base Software Elements in the Test Environments

Add or delete Software Elements as appropriate. If necessary, specify software patches referenced and/or required here.

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

Table : Software Elements

| Software Element Name | Version | Type and Other Notes |
| --- | --- | --- |
| NT Workstation |  | Operating System |
| Windows 2000 |  | Operating System |
| Internet Explorer |  | Internet Browser |
| Netscape Navigator |  | Internet Browser |
| MS Outlook |  | Email Client software |
| Network Associates McAfee Virus Checker |  | Virus Detection and Recovery Software |
|  |  |  |

Remove blank rows.

## Staffing and Training Needs

Table 8 describes the personnel resources needed to plan, prepare, and execute this Master Test Plan.

Table : Staffing Resources

| Testing Task | Quantity of Personnel Needed | Test Process | Duration/ Days |
| --- | --- | --- | --- |
| Create the Master Test Plan |  | Test Preparation | xxx days |
| Establish the Test Environment |  | Test Preparation | xxx days |
| Perform System Tests |  | Product Build  | xxx days |
| Etc. |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Identify training options for providing necessary skills and the estimated number of hours necessary to complete the training.

Remove blank rows.

Table 9 lists the personnel that require training.

Table : Training Needs

| Name | Training Need | Training Option | Estimated Training Hours |
| --- | --- | --- | --- |
| Alice Johnson | IBM Rational Robot ® | Attend IBM Rational Robot ® training  | 10 hrs. |
| Bill Smith | IBM Rational ClearQuest ® | Obtain IBM Rational ClearQuest ® training  | 4 hours |
|  |  |  |  |

Remove blank rows.

## Risks and Constraints

The Test Preparation process requires the performance of a risk assessment for test execution. Risks associated with the testing project are potential problems/events that may cause damage to the software, systems, patient, personnel, operating systems, schedule, scope, budget or resources. The risks, listed in the risk log, may impact scope and schedule, necessitating a deviation from this Master Test Plan.

The risk log was taken into consideration in the development of this test plan.

The risks identified in this Master Test Plan can be found in the risk log and may be recorded and tracked in an automated tool, such as, IBM Rational ClearQuest®.

## 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

## Attachment A – Approval Signatures

The Master Test Plan documents the project’s overall approach to testing and includes:

Items to be tested

* Test strategy
* Test criteria
* Test deliverables
* Test schedule
* Test environments
* Staffing and training needs
* Risks and constraints
* Test Metrics

This section is used to document the approval of the Master Test Plan during the Formal Review. The review should be ideally conducted face to face where signatures can be obtained ‘live’ during the review however the following forms of approval are acceptable:

1. Physical signatures obtained face to face or via fax
2. Digital signatures tied cryptographically to the signer
3. /es/ in the signature block provided that a separate digitally signed e-mail indicating the signer’s approval is provided and kept with the document.

NOTE: Delete the entire section above prior to final submission.

REVIEW DATE: <Date>

Signed: Date:

< Program/Project Manager >

Signed: Date:

< Business Sponsor Representative >

Signed: Date:

<Project Team Test Manager>

## 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 testing occurs when you simulate the number of users in order to stress an application's hardware and/or network infrastructure. Capacity testing is done to determine the capacity (CPU, Data Storage, LAN, WAN, etc.) of the system and/or network under test. |
| Compliance Testing | A type of testing that verifies that a collection of software and hardware fulfills given specifications. For example, these tests will minimally include: “core specifications for rehosting – ver.1.5-draft 3.doc”, Section 508 of The Rehabilitation Act Amendments of 1998, Race and Ethnicity Test, and VA Directive 6102 Compliance. It does not exclude any other tests that may also come up. |
| Component Integration Testing | Testing performed to expose defects in the interfaces and interaction between integrated components as well as verifying installation instructions. |
| Configuration Testing | A type of testing concerned with checking the programs compatibility with as many possible configurations of hardware and system software. In most production environments, the particular hardware specifications for the client workstations, network connections, and database servers vary. Client workstations may have different software loaded, for example, applications, drivers, and so on hand, at any one time; many different combinations may be active using different resources. The goal of the configuration test is finding a hardware combination that should be, but is not, compatible with the program. |
| Contention Testing | A type of performance testing that executes tests that cause the application to fail with regard to actual or simulated concurrency. Contention testing identifies failures associated with locking, deadlock, livelock, starvation, race conditions, priority inversion, data loss, loss of memory, and lack of thread safety in shared software components or data.  |
| Data and Database Integrity Testing | A type of testing that verifies that data is being stored by the system in a manner where the data is not compromised by the initial storage, updating, restoration, or retrieval processing. This type of testing is intended to uncover design flaws that may result in data corruption, unauthorized data access, lack of data integrity across multiple tables, and lack of adequate transaction performance. The databases, data files, and the database or data file processes should be tested as a subsystem within the application.  |
| Documentation Testing | Documentation testing is a type of testing that should validate the information contained within the software documentation set for the following qualities: compliance to accepted standards and conventions, accuracy, completeness, and usability. The documentation testing should verify that all of the required information is provided in order for the appropriate user to be able to properly install, implement, operate, and maintain the software application. The current VistA documentation set can consist of any of the following manual types:Release Notes, Installation Guide, User Manuals, Technical Manual, and Security Guide. |
| Error Analysis Testing | This type of testing verifies that the application checks for input, detects invalid data, and prevents invalid data from being entered into the application. This type of testing also includes the verification of error logs and error messages that are displayed to the user. |
| Exploratory Testing | A technique for testing computer software that requires minimal planning and tolerates limited documentation for the target-of-test in advance of test execution, relying on the skill and knowledge of the tester and feedback from test results to guide the ongoing test effort. Exploratory testing is often conducted in short sessions in which feedback gained from one session is used to dynamically plan subsequent sessions. |
| Failover Testing | A type of testing test that ensures an alternate or backup system properly “takes over” (i.e., a backup system functions when the primary system fails). Failover Testing also tests that a system continually runs when the failover occurs, and that the failover happens without any loss of data or transactions. Failover Testing should be combined with Recovery Testing. |
| Installation Testing | A type of testing that verifies that the application or system installs as intended on different hardware and software configurations, and under different conditions (e.g., a new installation, an upgrade, and a complete or custom installation). Installation testing may also measure the ease with which an application or system can be successfully installed, typically measured in terms of the average amount of person-hours required for a trained operator or hardware engineer to perform the installation. Part of this installation test is to perform an uninstall. As a result of this uninstall, the system, application and database should return to the state prior to the install. |
| Integration Testing | An incremental series of tests of combinations or sub-assemblies of selected components in an overall system. Integration testing is incremental in a successively larger and more complex combinations of components tested in sequence, proceeding from the unit level (0% integration) to eventually the full system test (100% integration).  |
| Load Testing | A performance test that subjects the system to varying workloads in order to measure and evaluate the performance behaviors and abilities of the system to continue to function properly under these different workloads. Load testing determines and ensures that the system functions properly beyond the expected maximum workload. Additionally, load testing evaluates the performance characteristics (e.g., response times, transaction rates, and other time-sensitive issues). |
| Migration Testing | A type of testing that follows standard VistA and 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 completedData tables are successfully createdParallel test for confirmation of data integrityReview output report, before and after migration, to confirm data integrityRun 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 the Health Insurance Portability and Accountability Act (HIPAA). |
| Product Component Testing | Product Component Testing (aka Unit Testing) is the internal technical and functional testing of a module/component of code. Product Component Testing verifies that the requirements defined in the detail design specification have been successfully applied to the module/component under test. |
| Recovery Testing | A type of testing that causes an application or system to fail in a controlled environment. Recovery processes are invoked while an application or system is monitored. Recovery testing verifies that application or system, and data recovery is achieved. Recovery Testing should be combined with Failover Testing. |
| Regression Test | A type of testing that validates existing functionality still performs as expected when new functionality is introduced into the system under test.  |
| Risk Based Testing | A type of testing based on a defined list of project risks. It is designed to explore and/or uncover potential system failures by using the list of risks to select and prioritize testing. |
| Section 508 Compliance Testing | A type of test that (1) ensures that persons with disabilities have access to and are able to interact with graphical user interfaces and (2) verifies that the application or system meets the specified Section 508 Compliance standards. |
| Security Testing | A type of test that validates the security requirements and to ensure readiness for the independent testing performed by the Security Assessment Team as used by the Assessment and Authorization Process. |
| Smoke Test | A type of testing that ensures that an application or system is stable enough to enter testing in the currently active test phase. It is usually a subset of the overall set of tests, preferably automated, that touches parts of the system in at least a cursory way.  |
| Stress Testing | A performance test implemented and executed to understand how a system fails due to conditions at the boundary, or outside of, the expected tolerances. This failure typically involves low resources or competition for resources. Low resource conditions reveal how the target-of-test fails that is not apparent under normal conditions. Other defects might result from competition for shared resources (e.g., database locks or network bandwidth), although some of these tests are usually addressed under functional and load testing. Stress Testing verifies the acceptability of the systems performance behavior when abnormal or extreme conditions are encountered (e.g., diminished resources or extremely high number of users). |
| System Testing | System testing is the testing of all parts of an integrated system, including interfaces to external systems. Both functional and structural types of testing are performed to verify that the system performance, operation and functionality are sound. End to end testing with all interfacing systems is the ultimate version.  |
| Usability Testing | Usability testing identifies problems in the ease-of-use and ease-of-learning of a product. Usability tests may focus upon, and are not limited to: human factors, aesthetics, consistency in the user interface, online and context-sensitive help, wizards and agents, user documentation.  |
| User Functionality Test | User Functionality Test (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 |
| --- | --- | --- | --- |
| September 2020 | 1.19 | Replaced all references to ProPath with Process Asset Library (PAL) and corrected broken links | Quality Continuous Improvement Organization (QCIO) |
| 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 |

Place latest revisions at top of table.

The Template Revision History pertains only to the format of the template. It does not apply to the content of the document or any changes or updates to the content of the document after distribution.

The Template Revision History can be removed at the discretion of the author of the document.

Remove blank rows.