Test Evaluation Summary

Genomic Information System for Integrated Science 2

(Genisis2) Technical Services

Release 3



June 2017

Document Version 1.0

Department of Veterans Affairs

Revision History

**Note**: The revision history cycle begins once changes or enhancements are requested after the Communications Plan has been baselined.

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| 06/16/2017 | 1.0 | Release 3 Test Evaluation Summary Report | Booz Allen Hamilton |

Artifact Rationale

The test evaluation document is the primary output of the test and evaluation process, an integral part of the systems engineering process, which identifies levels of performance and assists the developer in correcting deficiencies.

The PMAS Directive cites the Service Delivery and Engineering (SDE) organization as having primary responsibilities for system testing and certification.

Table of Contents

[1. Test Evaluation Introduction 4](#_Toc485409661)

[1.1. Test Evaluation Scope 4](#_Toc485409662)

[1.2. Test Architecture 5](#_Toc485409663)

[1.3. Test Environment/ Configuration 6](#_Toc485409664)

[1.4. Installation Process 8](#_Toc485409665)

[2. Test Data 9](#_Toc485409666)

[3. Issues 9](#_Toc485409667)

[4. Test Execution Log 9](#_Toc485409668)

[5. Test Results Summary 12](#_Toc485409669)

[5.1. Defect Severity and Priority Levels 12](#_Toc485409670)

[5.2. Total Defects by Severity Level 12](#_Toc485409671)

[5.3. Defects by Severity Level and Increment 12](#_Toc485409672)

[5.4. Performance Testing 12](#_Toc485409673)

[5.4.1. Test Event 12](#_Toc485409674)

[5.4.2. Requirements Coverage Status 14](#_Toc485409675)

[5.4.3. Test Design 14](#_Toc485409676)

[5.4.4. Performance Test Results 14](#_Toc485409677)

[5.4.5. Transaction Response Times 14](#_Toc485409678)

[6. Test Coverage 14](#_Toc485409679)

[6.1. Requirements Covered 14](#_Toc485409680)

[6.2. Section 508 Compliance Coverage 14](#_Toc485409681)

[7. Suggested Actions 15](#_Toc485409682)

[8. Defect Severity and Priority Definitions 15](#_Toc485409683)

[8.1. Defect Severity Level 15](#_Toc485409684)

[8.1.1. Severity Level 1 – Critical 15](#_Toc485409685)

[8.1.2. Severity Level 2 - High 15](#_Toc485409686)

[8.1.3. Severity Level 3 - Medium 16](#_Toc485409687)

[8.1.4. Severity Level 4 - Low 16](#_Toc485409688)

[8.2. Priority Classifications 16](#_Toc485409689)

[8.2.1. Priority 1 - Resolve Immediately 16](#_Toc485409690)

[8.2.2. Priority 2 - Give High Attention 17](#_Toc485409691)

[8.2.3. Priority 3 - Normal Queue 17](#_Toc485409692)

[8.2.4. Priority 4 - Low Priority 17](#_Toc485409693)

[9. Optional Tables, Charts, and Graphs 18](#_Toc485409694)

[10. Document Approval Signatures 19](#_Toc485409695)

# Test Evaluation Introduction

The Genomic Information System for Integrated Science 2 (Genisis2) provides additional functionality to Genisis 1.0 platform through automating data processing, data request transactions, data request tracking for integrating VA Informatics and Computing Infrastructure (VINCI) clinical data into Genisis2. In addition, to providing enhanced system administration. The purpose of Genisis2 testing is to measure the functional quality of the software against the specified user requirements, prior to the system’s release into the production environment. The testing process covers the major business functions for creations of data requests to support human research projects. It also provides the necessary traceability to the design and requirements artifacts to ensure that all components have been fully developed and can eventually operate in a production environment.

The Test Evaluation Summary collects, organizes, and presents the test results and key measures of testing to enable objective quality evaluation and assessment. This report provides the system’s stakeholders with an assessment of the adequacy of the management, operational, and technical controls used to protect the confidentiality, integrity, and availability of the system and the data it stores, transmits or processes.

## Test Evaluation Scope

The Genisis2 Test Result Report is for the Department of Veterans Affairs (VA) Office of Research and Development (ORD). The scope of the report includes the assessment of the system level management, operational, and technical controls as documented in the System Design Document (SDD) and Master Test Plan that directly support the system.

Testing provides the chance to validate the functionality of Genisis2 Release 3, which includes the following high-level functional capabilities:

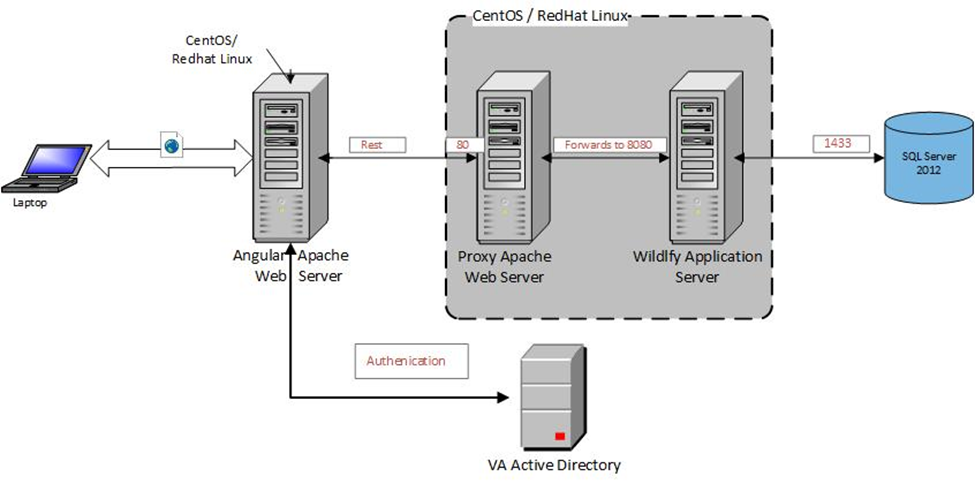
| **Role / Work Stream** | **Features / Functionality** |
| --- | --- |
| **Requestor (UI)** | * Regression Create/Modify/Track Data Request * Automatic email notification for status changes * Add Comments for Return only |
| **Data Destination Manager (UI)** | * Regression Create/ Modify/Track/Approve/Deny Return Data Request/Results Accepted/Results Not Accepted * Automatic email notification for status changes * Add Comments |
| **Data Source Manager (UI)** | * Regression Request Accepted or Request Not Accepted (on behalf of VINCI) * Add Comments and Deliver Results |
| **History (UI)** | * Verify High Level and Detailed History is captured for each transaction related to each request |
| **Comments (UI)** | * Provide the ability to add comments throughout the lifecycle of a data request |
| **Backend** | * Data File Assessment (file checking) * COPY Table Function (VINCI to Genisis2) |
| **Security** | * Integration with VA’s Active Directory * Role-based access with two-factor authentication |

## Test Architecture

Genisis2 Sustainment testing is conducted in a Software Quality Assurance (SQA) environment which is located at the Austin Information Technology Center (AITC). As shown in Figure 1, the testers connect to the Genisis2 Solution via standard Web Application ports over HTTPS. All communication between the Genisis2 components is through standard, VA approved communication protocols. Please refer to Section 1.3 for more details about the components that are part of the environment.

Figure 1: Environment Architecture

VA Intranet



The Test Server Architecture consists of the following components:

1. Angular Apache Webserver running on RHEL
2. Wildfly Application Server running on RHEL
3. Microsoft SQL Server Database running on Microsoft Windows 2008 R2

The Webserver consists of Apache running on Red Hat Enterprise Linux (RHEL). It employs Angular JS as the User Interface (UI) framework.

The Application Server runs on Red Hat Enterprise Linux (RHEL). It supports two main components – The Application Code in Java and the Java Business Process Management (JBPM) engine.

The Java code handles the application logic and serves as the backend engine that the webserver calls using the API interface. It also manages the user interactions of the workflow management. The Java code also uses the Spring framework and manages the Object-Relational mapping needed for the application.

The Java Business Process Management engine captures the workflow logic and executes it. The JBPM engine manages its database entries in a separate database hosted in the common database server instance of Microsoft SQL server. Changes to the workflow can be made in this engine and deployed.

The Database Server is a Microsoft Windows 2008 R2 server and hosts a Microsoft SQL server 2012 database engine. This database holds and manages all the data related to the application. The JBPM engine has a separate database here in the same instance and manages workflow related data in it.

## Test Environment/ Configuration

The Genisis2 Test environment is divided into the following conceptual layers and the related physical layer components depicted in Figure 1:

* Presentation
* Application
* Data

Authentication and Authorization is not a separate layer of the application, but is necessary to describe it when outlining how the application works.

The Presentation layer is the top layer of the Genisis2 Solution architecture and represents the way a user interacts with the Genisis2 solution. It uses Apache Web Server software and Angular JS as the User Interface Framework.

The Application layer uses a combination of Java Code and Java Business Process Management engine to capture an execute business processes and rules. The Presentation layer calls the Application layer, which is the backend application engine using an Application Programming Interface (API).

The security model protects the platform from unauthorized access across the web. Applications that interface with the Genisis2 solution platform use standard VA standardized web services to communicate. The Platform also controls access to data through security, controls access to the database and raises events for workflow processes and custom business logic implementations.

The Data layer, the lowest layer in the architecture, provides the centralized databases for storing the Genisis2 Solution data. The Data layer also consists of the Document Repository, which serves content management and dashboard capabilities, as well as, workflow management used by the platform layer.

Figure 2: Security Architecture

Genisis2 Security Architecture



**Authentication and Authorization**

Figure 2 illustrates the Genisis2 security architecture. User Names and Passwords are controlled by centralized VA LDAP access control processes. Password Expiry and other administrative processes are controlled by the VA LDAP group.

Genisis2 uses its own local database server to handle Genisis2 User Roles. Genisis2 user types are managed within the local database server to provide the user some functions and hide others from them. For example, a user is recognized as a Requestor, Data Destination Manager or Genisis2 System Administrator and are afforded different levels of functionality within the system.

When a new user is created within Genisis2 it needs to authenticate that person’s username and Research groups they belong from the VA LDAP server and update the local database.

Genisis2 authenticates any user with only the local database server and at the same time, information about which Research Groups they belong to, which projects they belong to, who their Data Destination Manager is all extracted from the server at this time.

Encryption is handled by https protocol with ssl certificates installed on the servers and VA LDAP servers natively.

**Identity Management**

Genisis2 supports the following user types:

1. Requestor
2. Data Destination Manager
3. Data Source Manager
4. Genisis2 System Administrator (planned for a future release)

The Requestor has the minimum set of activities they can access and perform in the application. For example, the Requestor can generate and track their own data requests.

The Data Destination Manager has all the capabilities of a Requestor, but can review and approve requests, review and approve data results, and track requests more broadly.

The Data Source Manager addresses any questions about the data that they may have with the Requestor (through the Data Destination Manager and not directly), prepares and places the data in a Source landing zone database, and notifies the Data Destination Manager of its location.

The Data Destination Manager will then copy the data over from the Source landing zone database to the Destination landing zone database, extract the data to a flat file, and perform any additional cleanup that may be required. Personally Identifiable Information (PII) and Protected Health Information (PHI) information is then removed from this data and copied over to the specific Study Mart set up for the Requestor.

The Genisis2 System Administrator is a super user that has access to all of the functions that the Requestor and the Data Destination Manager have; and in addition, the Genisis2 System Administrator can Create, Modify, or Delete users. The Genisis2 System Administrator is responsible for creating a user account within Genisis2 and assigning one of the roles. The Genisis2 application then coordinates with the VA LDAP server to create and store the role information.

**Access Control**

User Names and Passwords are controlled by centralized VA LDAP access control processes. Password Expiry and other administrative processes are controlled by the VA LDAP group.

Genisis2 coordinates with the local VA LDAP server for username and password authentication. Genisis2 user types are managed within a Genisis2 database server to provide the user some functions and hide others from them. For example, a user is recognized as a Requestor, Data Destination Manager, Data Source Manager, or Genisis2 System Administrator, and are afforded different levels of functionality within the application.

## Installation Process

The Genisis2 Development Team performed all required installations in the VA environment. For details, please see sections 1.2 and 1.3.

# Test Data

All the test cases executed for Genisis2 Release 3 testing are designed to create and manage data requests as part of the execution of the test case. There are also several scenarios that execute the copying of data from one source database to a destination database with a verification by accessing the database log.

# Issues

There are no issues to report for this project.

# Test Execution Log

The Test Execution Log is used to record the manual execution of test scripts and to document the test results for each test script. The table below shows the different testing iterations that took place during the software development lifecycle. Software Quality Assurance (SQA) testing and User Acceptance Testing (UAT) testing was performed with all of the test cases used by the Genisis2 Test Team.

The Test Execution Log includes the following and is located in Rational

|  |  |
| --- | --- |
| **Item** | **Definition** |
| **Test Case** | The numerical identifier for the Test Case associated with this Test Case. |
| **Test ID** | The identification number assigned to this test case. |
| **Date Tested** | The date the test was executed. |
| **User Story** | The name and number of the user story associated with the test case. |
| **Requirement** | The name and number of the requirement associated with the test case. |
| **Actual Result:**  **Pass/Fail** | The test outcome: P=Pass, F=Fail. |
| **Tester Name** | The name of the tester that performed test execution. |

The table below provides a list of the test cases executed their associated requirements and results for Genisis2 Release 3:

| **Test Case #** | **Requirement** | **Pass/Fail** | **Date** |
| --- | --- | --- | --- |
| TC0063 Data Source Manager\_Review Details\_ of Approved\_Data Request | 4.4.1 The system shall provide the ability for a Data Source Manager to review the approved data request. | P | 6/5/2017 |
| TC0064 Data Source Manager\_2Factor\_Login | 4.4.1.1: The system shall require the Data Source Manager to log in using 2-factor authentication. | P | 6/5/2017 |
| TC0065Data Source Manager\_Display\_Request View\_Icons | 4.4.1.2: The system shall allow the Data Source Manager to view a list of data requests that are pending acceptance. | P | 6/5/2017 |
| TC0066 Data Source Manager\_Review\_Actions and History View on Requests | 4.4.1.3: The system shall allow the Data Source Manager to select a data request and view the details. | P | 6/5/2017 |
| TC0067 Data Source Manager \_read only fields\_except for comments | 4.4.1.4: The system shall display the request as read-only. | P | 6/5/2017 |
| TC0068 Data Source Manager\_Manage\_Data Requests | 4.4.2: The system shall provide the ability for a Data Source Manager to manage a data request. | P | 6/5/2017 |
| TC0069 Data Source Manager\_Add Comments\_ Changing Status to Request Cannot be Fulfilled | 4.4.2.1: The system shall allow the Data Source Manager to reject the data request. The system sets the status to Request Cannot Be Fulfilled. | P | 6/5/2017 |
| TC0070 Data Source Manager\_Add Comments \_Changing Status\_to Accepted | 4.4.2.2: The system shall allow the Data Source Manager to accept the data request. The system shall set the status to Request Accepted. | P | 6/5/2017 |
| TC0071 Data Source Manager\_Add Comments to Request\_No Status Change | 4.4.2.3: The system shall allow the Data Source Manager to add a Comment to the data request without changing the status of the request. | P | 6/5/2017 |
| TC0072 Data Source Manager\_Verify email notifications for Action\_Request Cannot Be Fulfilled | 4.4.2.4: The system shall generate an email notifications to the Requester and Data Destination Manager when the request has been rejected by the Data Source Manager. | P | 6/5/2017 |
| TC0073 Data Source Manager\_Verify Email notifications for Action\_Request Accepted | 4.4.2.5: The system shall generate an email notification to the Requester and Data Destination Manager when the request has been accepted by the Data Source Manager. | P | 6/5/2017 |
| TC0074 Data\_Destination\_Manager\_has\_ability\_to\_CopyTable | 4.5.1: The system shall provide the ability for a Data Destination Manager to Copy tables from a Source location to a Destination location. | P | 6/5/2017 |
| TC0075 System\_display\_CopyTable\_button\_for\_Results\_Delivered | 4.5.1.1: The system shall display the Copy Table button on the Manage Request screen  for a request that has been delivered (Status = Results Delivered). | P | 6/5/2017 |
| TC0076 DDM\_enter\_TableNames\_field | 4.5.1.2: The system shall display a Table Name dialog box to capture 1 table name to be copied. | P | 6/5/2017 |
| TC0077 System\_displays\_CopyTable\_on\_Manage\_Request\_Page | 4.5.1.3: The system shall display the copy table fields when the Copy Table button of the Manage Request screen is executed. | P | 6/5/2017 |
| TC0078 System\_notify\_Source\_Schema\_incorrect | 4.5.1.3a: The system shall return a notification when the Source schema name is incorrect. | P | 6/5/2017 |
| TC0079 System\_notify\_Destination\_Schema\_incorrect | 4.5.1.3b: The system shall return a notification when the Destination schema name is incorrect. | P | 6/5/2017 |
| TC0080 System\_notify\_TableName\_incorrect | 4.5.1.3c: The system shall return a notification when the Table name is incorrect. | P | 6/5/2017 |
| TC0081 System\_notify\_Copy\_successfully | 4.5.1.3d: The system notifies the Data Destination Manager that the copy process completed successfully. | P | 6/5/2017 |
| TC0082 System\_display\_Copy\_fields\_when\_executed | 4.5.1.4: The system shall display the copy fields when the Copy button is executed. | P | 6/5/2017 |
| TC0083 - Display information related to deliver results | 4.6.1. The system shall display the ability for a Data Source Manager to deliver the location of the request data. | P | 6/5/2017 |
| TC0084 - Deliver Results button available for DSM role only | 4.6.1.1. The system shall display the Deliver Results button on a data request in the Request Accepted status. | P | 6/5/2017 |
| TC0085 - Comments required to Deliver Results. | 4.6.1.2. The system shall require the comment when the Deliver Request button is executed. | P | 6/5/2017 |
| TC0086 - DSM reciept of email notificaion that data results are ready. | 4.6.1.3. The system shall send an email notification to the assigned Data Destination Manager that the results are ready for review. | P | 6/5/2017 |
| TC0087 - Requestor reciept of email notificaion that data results are ready. | 4.6.1.4. The system shall send an email notification to the Requester that the results were delivered to the Data Destination Manager. | P | 6/5/2017 |

Issues reported during the test execution process were entered into Rational Team Concert (CM), a work item tracking system. Description of each work item was provided along with the test cycle in which it was found (Unit vs. SQA vs. UAT). Defects were either associated with a test case that failed or were judged to be a general defect of the application (e.g., a GUI/user interface issue). Severity level and Priority level were assigned to each defect along with a unique defect code generated after entering it into Rational. Severity levels and Priority levels were determined using the standard VA definitions that were documented in a tab of the defect tracking spreadsheet and shown in Section 5 of this report. The date of when each defect was opened or closed was also provided, as well as the status of the defect.

# Test Results Summary

## Defect Severity and Priority Levels

A defect is defined as a flaw in a component or system that can cause the component or system to fail to perform its required function (e.g., an incorrect statement or data definition. A defect, if encountered during execution, may cause a failure of the component or system).

Defects are categorized according to severity and priority levels. The test analyst assigns the severity, while the development manager assigns the priority for repair. For more information, see Defect Severity and Priority Definition in this Test Evaluation.

## Total Defects by Severity Level

At the end of Genisis2 Release 3 testing, including SQA and UAT, there was one (1) outstanding minor defect reported. Feedback/comments captured during the test effort were categorized. The open defect will be corrected in Build/Release 4, as approved by the VA or during the sustainment period following deployment to production.

## Defects by Severity Level and Increment

The defect identified during this test effort was categorized as minor and changes to this defect will be implemented in Release 4 or in a future sustainment release.

## Performance Testing

This section is not applicable for Genisis2 Release 3 based on the limited number of initial users. Performance testing is a scheduled requirement for Release 4.

### Test Event

Genisis2 Release 3 testing started May 24, 2017. The Genisis2 Test Team created 25 test cases to support this effort. During the internal test, the testers executed all test cases within the spreadsheet applicable to Release 3 and executed regression tests to insure there was no impact to Release 1 and Release 2 functionality.

Once the case was executed and results analyzed, the spreadsheet was marked the step as either Pass (P) or Fail (F).

The executed scripts were tabulated by test status: Passed, or Failed.

The Genisis2 Test Team captured test results and triage was conducted internally with the development team to validate and classify the results.

The summary test metrics are shown below:

* 25 test cases were executed
* 13 test cases Passed (by the end of Sprint 4)
* 13 test cases had defects filed against them
* 1 test defect remains to be corrected in Release 4 or a future sustainment release

For a full list of defects found during development and internal testing, please see the attached JIRA Report.



The Test Cases were submitted to the VA-SQA Team for review and use as the basis for the SQA Test. The Internal Test Team scheduled a session to perform test execution for all the test scenarios applicable to Release 3 as well regression test Release 1 and Release 2 functionality on May 30, 2017. The VA-SQA Team executed all scenarios with the exception of the ‘Copy Table’ function during the first session. No issues were found and logged in Rational Team Concert (CM). A second session was held June 13, 2017 to demonstrate the copy table scenarios once the changes were in place.

The summary test metrics are shown below:

* 25 test cases were executed
* 25 test cases Passed

For details about the three defects, please see the Genisis2 Rational Instance (CM).

<https://DNS/ccm/web/projects/Genisis%20(CM)#action=com.ibm.team.apt.viewPlan&page=com.ibm.team.apt.web.ui.plannedItems&id=_nEFWEGisEeaGZ_dHulLUfA&planMode=com.ibm.team.apt.viewmodes.internal.iteration>

For more details about the test that were executed for SQA, please see the attached spreadsheet.



For details about the defect, please see the Genisis2 Rational Instance (CM).

The Business Owner identified three UAT testers for Genisis2 Release 3. A kickoff meeting was held to obtain preliminary user information to and identify user roles prior to the UAT. The Genisis2 SME modified the spreadsheet to consolidate the scenarios into one step-by-step scenario and the Internal Test Team added columns and fields to include tester information, user/role data and pass/fail information. The Internal Test Team scheduled one session, June 12, 2017 to facilitate test execution for all of the test scenarios applicable to Release 3 and completed the test without any defects or issues identified.

The summary test metrics are shown below:

* 25 test cases were executed
* 25 test cases Passed

For more details about the test that were executed for UAT, please see the attached spreadsheet.



### Requirements Coverage Status

The requirements coverage is established within the Rational Instance via linkage between the requirements in RM and the test cases in RQM.

### Test Design

Test design is not applicable for Release 3.

### Performance Test Results

Performance test results are not applicable for Release 3.

### Transaction Response Times

Transaction response times are not applicable for Release 3.

# Test Coverage

## Requirements Covered

Please refer to the Requirements Traceability Matrix (RTM) within Rational for a detailed mapping between requirements and test cases. The Internal Test Team established coverage by linking the requirements in RM with the test cases in RQM.

## Section 508 Compliance Coverage

The Genisis2 Team has completed internal verification of the code to conform to Section 508 Compliance standards by using the VA-approved tool, JAWS. Our internal 508 SME evaluated the software resulting in an internal pass for the Release 3 code. The report from that evaluation has been included within this document for reference. The documentation required for self-certification and request for audit was submitted to the VA-508 Team on Friday, June 9, 2017.



# Suggested Actions

All comments made by SQA and/or UAT participants have been collated and updated where required within the test spreadsheets. Please see the documents attached in Section 5.4.1.

# Defect Severity and Priority Definitions

The classification of defects within a system examines both the severity and priority of the defect.

Severity is a measure of how great the impact is on the user’s ability to complete the documented actions within the system.

Priority determines the speed with which a given defect must be repaired.

Defect classification may be determined either because testing is delayed by a failure in the system or because a cumbersome workaround prevents a user from completing the assigned tasks. Both severity and priority measures must be recorded when scheduling defect resolution tasks.

## Defect Severity Level

The following subsections identify the defect severity levels.

### Severity Level 1 – Critical

Institute of Electrical and Electronics Engineers (IEEE) definition: The defect results in the failure of the complete software system, of a subsystem, or of a software unit (program or module) within the system.

* Any defect that compromises patient safety or system security. Examples of system security defects include breach of confidentiality requirements of the Privacy Act, the Health Insurance Portability and Accountability Act (HIPAA), or Federal Tax Information guidelines.
* Loss of system functionality critical to user operations with no suitable workaround, i.e., there is no way to achieve the expected results using the application.
* System crash or hang that prevents further testing or operation of the complete application or a section of the application.
* Any defect that causes corruption of data from a result of the system (as opposed to user error).
* Any defect in which inappropriate transmissions are consistently generated or appropriate transmissions of HL7 messages fail to be generated.
* Loss of functionality resulting in erroneous eligibility/enrollment determinations or communications not being sent.

### Severity Level 2 - High

IEEE definition: The defect results in the failure of the complete software system, of a subsystem, or of a software unit (program or module) within the system. There is no way to make the failed component(s) function. However, there are acceptable processing alternatives which will yield the desired result.

* A major defect in the functionality that does not result in corruption of data.
* A major defect in the functionality resulting in a failure of all or part of the application, where:
* The expected results can temporarily be achieved by alternate means. The customer indicates the work around is acceptable for the short term.
* Any defect that does not conform to Section 508 standards.
* Any defect that results in inaccurate or missing requirements.
* Any defect that results in invalid authentication or authentication of an invalid end user.

### Severity Level 3 - Medium

IEEE definition: The defect does not result in a failure, but causes the system to produce incorrect, incomplete, or inconsistent results, or the defect impairs the systems usability.

* Minor functionality is not working as intended and a workaround exists but is not suitable for long term use
* The inability of a valid user to access the system consistent with granted privileges
* Typographical or grammatical errors in the application, including installation guides, user guides, training manuals, and design documents
* Any defect producing cryptic, incorrect, or inappropriate error messages
* Any defect that results from the use of non-standard data terminology in the application or documentation, as defined by the Department of Veterans Affairs
* Cosmetic issues that are important to the integrity of the product, but do not result in data entry and or data quality problems.

### Severity Level 4 - Low

IEEE definition: The defect does not cause a failure, does not impair usability, and the desired processing results are easily obtained by working around the defect.

* Minor loss of, or defect in the functionality where a long-term use exists
* Low-level cosmetic issues.

## Priority Classifications

The following subsections identify the appropriate actions for defects at each priority level, per definitions of IEEE.

### Priority 1 - Resolve Immediately

Further development and/or testing cannot occur until the defect has been repaired. The system cannot be used until the repair has been affected.

### Priority 2 - Give High Attention

The defect must be resolved as soon as possible because it is impairing development and/or testing activities. System use will be severely affected until the defect is fixed.

### Priority 3 - Normal Queue

The defect should be resolved in the normal course of development activities. It can wait until a new build or version is created.

### Priority 4 - Low Priority

The defect is an irritant that should be repaired, but can be repaired after more serious defects have been corrected.

# Optional Tables, Charts, and Graphs

Optional tables, charts and graphs are not needed at this time.

# Document Approval Signatures

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

Program/Project Manager Date

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

Business Sponsor Representative Date

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

Lead Tester Date