Perceptive Reach

Integrated Reach Database System

(IRDS)

Master Test Plan



Department of Veterans Affairs

**September 2016**

**Version 4.0**

Revision History

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| 9/13/2016 | 4.0 | Management Review | Matthew Robinson |
| 9/12/2016 | 3.9.1 | Quality Assurance Review with minor updates throughout. | Kaitlin Reskovac |
| 9/9/2016 | 3.9 | Updated the title page and page headers to reflect the correct month. | Kaitlin Reskovac |
| 5/25/2016 | 3.8 | Management Review | Matthew Robinson |
| 5/17/2016 | 3.7 | Reviewed the document. | Vasudha Upadhyaya |
| 5/16/2016 | 3.6 | Updated table in section “5.7 Productivity & Support Tools”  Added sections “5.3. Stress Testing”, “5.2. Break Testing”, “5.4. Corner Case Testing”, and “5.5. Performance Testing”  Updated Table 9 in section “8.1. Base Software Elements in the Test Environments”  Added ticket PR-3860 to section “3.1. Overview of Test Inclusions” | Kaitlin Reskovac |
| 5/10/2016 | 3.5 | Updated the title page and page headers to reflect the correct month  Added section “5.2 Stress and Break Testing” | Kaitlin Reskovac |
| 4/11/2016 | 3.4 | Updated requirements in section “3.1. Overview of Test Inclusions”  Updated section “4.3. Pilot Testing” | Kaitlin Reskovac |
| 4/8/2016 | 3.3 | Updated title page and page headers to reflect the correct month  Updated section “4.3. Pilot Testing” to include information about WinCollect Testing | Kaitlin Reskovac |
| 3/30/2016 | 3.2 | Updated title page and page headers to reflect correct month  Updated section “3.2. Overview of Test Exclusions” | Kaitlin Reskovac |
| 2/19/2016 | 3.1 | Updated section “4.3. Pilot Testing” to include test plans executed in the option year | Kaitlin Reskovac |
| 9/14/2015 | 3.0 | Final Review | Monica Mohler / Paul Bradley |
| 9/13/2015 | 2.4 | Formatting updates and peer review comment resolution | Matthew Robinson / Kaitlin Reskovac |
| 9/13/2015 | 2.3 | QA and Compliance Check | Monica Mohler / Paul Bradley |
| 9/4/2015 | 2.2 | September Updates & Peer Review | Kaitlin Reskovac |
| 8/7/2015 | 2.1 | August Updates | Kaitlin Reskovac |
| 7/24/2015 | 2.0 | July Updates | Kaitlin Reskovac |
| 6/5/2015 | 1.9 | Final Review | Monica Mohler / Paul Bradley |
| 6/4/2015 | 1.8 | Peer Review | Radina Ivanova |
| 6/3/2015 | 1.7 | June Updates | Andrew Smith / Chandrakala |
| 5/11/2015 | 1.6 | Final Review | Monica Mohler / Paul Bradley |
| 5/4/2015 | 1.5 | Peer Review | Radina Ivanova |
| 5/1/2015 | 1.4 | May Updates | Andrew Smith / Chandrakala |
| 2/5/2015 | 1.3 | Final Review | Monica Mohler / Paul Bradley |
| 2/5/2015 | 1.2 | Peer Review | Radina Ivanova |
| 2/5/2015 | 1.1 | Updates | Andrew Smith |
| 12/11/2014 | 1.0 | Final Review | Monica Mohler / Paul Bradley |
| 12/10/2014 | .02 | Peer Review | Radina Ivanova |
| 11/25/2014 | .01 | Initial Version | Chandrakala Gattu |

Table of Contents

1. Introduction 6

1.1. Background 6

1.2. Purpose 7

1.3. Test Objectives 7

2. Testing Approach 8

2.1. TDD 8

2.1.1. Processes 8

3. Items to Be Tested 11

3.1. Overview of Test Inclusions 11

3.2. Overview of Test Exclusions 21

3.3. Product Component Test 21

3.4. Component Integration Test 21

3.5. System Tests 21

4. Automated Testing 22

4.1. Regression Testing 22

4.2. Feature File Example 22

4.3. End User Testing 22

4.4. Pilot Testing 23

4.5. User Functional Testing 32

5. Testing Techniques 33

5.1. Risk-based Testing 33

5.2. Break Testing 33

5.3. Stress Testing 33

5.4. Corner Case Testing 34

5.5. Performance Testing 34

5.6. Enterprise Testing 35

5.6.1. Security Testing 35

5.6.2. Privacy Testing 35

5.6.3. Section 508 Compliance Testing 35

5.7. Productivity and Support Tools 38

6. Criteria 40

6.1. Pass/Fail Criteria 40

6.2. Acceptance Criteria 41

7. Test Deliverables 41

8. Test Environment 42

8.1. Base Software Elements in the Test Environments 43

9. Test Metrics 43

10. Acronyms 43

10.1. References 45

10.1.1. Documentation, Resources, and Repositories 45

Appendix A 47

A.1 Office of Cyber Security 47

# Introduction

This Master Test Plan (MTP) is developed for the Perceptive Reach Integrated Reach Database System (IRDS) project for the Department of Veterans Affairs (VA). VA is seeking to expand suicide prevention to include upstream approaches designed to reduce initiation or escalation of a suicide risk factor. Upstream suicide interventions target individuals or groups who exhibit biological, psychological, or social risk factors that are more prominent among high-risk groups than among the larger population. Understanding the unique needs of our nation’s Veterans and the military culture as it relates to stigma and mental health is important for early intervention. The goal of the IRDS innovation is to promote the general health of the Veteran population and effectively intervene in issues before they escalate into crisis.

The IRDS solution will harness the power of large and diverse data stores to aggregate, analyze and identify risk onset as well as reveal previously unidentified at-risk individuals and populations as a holistic and integrated approach. The IRDS innovation will serve to bolster the three major components of Veterans Health Administration’s (VHA) Strategic Plan for Suicide Prevention: surveillance, risk and protective factors, and prevention interventions. The IRDS innovation will target antecedent events specific to Veteran populations prior to the onset of risk and its mitigation.

## Background

The IRDS development and field pilot combines technology, outreach and clinical support to realize a clinically based data-driven early intervention and treatment solution aimed at suicide prevention. The application will include capability for analyzing multiple and integrated data sets with cutting-edge data analytic techniques and visualizations to identify at-risk individuals and populations and provide proactive and secure notifications of these results to Veteran support services.

A central component of the IRDS project is VA’s Suicide Data Repository (SDR). The SDR is a centralized SQL Server database consolidating multiple sources of data containing suicide and mortality data of Veterans. The IRDS project proposes to expand the capabilities of the SDR to include new interfaces to clinical data sources, integrated data analytics/predictive modelling capabilities, a surveillance dashboard, and secure messaging – thus enhancing SDR capabilities to include upstream suicide intervention.

IRDS will be an integrated system that builds upon the existing SDR with the following capabilities:

* **Reach Database.** A SQL database used to aggregate new data sources and relevant SDR data.
* **Data Analytics Platform and Dashboard.** An integrated collection of analytics and visualization tools, including a surveillance dashboard aimed at identifying at-risk individuals and populations.
* **Direct Messaging.** A method to construct and transmit a secure message to authorized outreach and intervention service providers.
* **Outreach and Intervention.** A pilot workflow that includes the process by which outreach and intervention resources are notified and act upon the data provided.

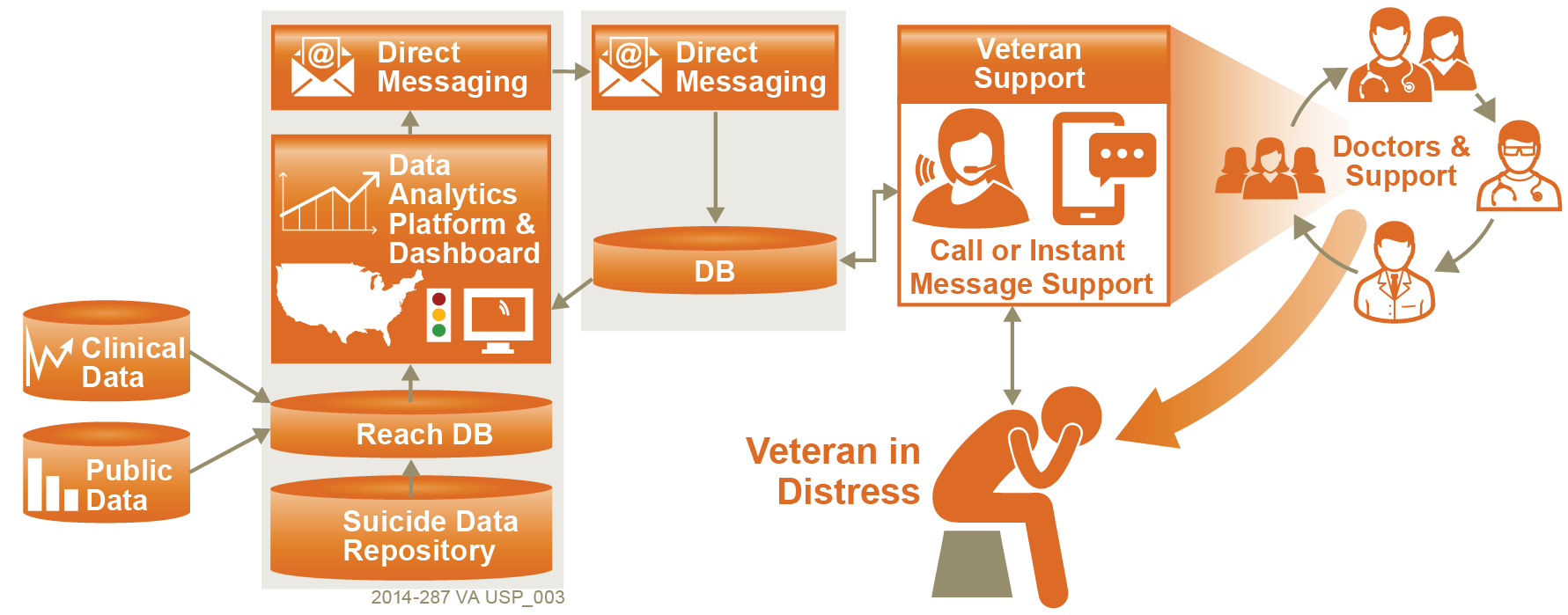


Figure 1: IRDS Architecture

## Purpose

The purpose of this MTP for the IRDS project is to provide an overview of the project and describe the testing approaches being used.

It documents how the project defines, implements, and assures quality during the prototype software development process. The test plan is a central artifact detailing the approaches taken to guarantee deliverable acceptance, along with the procedures and activities used to govern the prototype meeting its defined requirements.

This plan is recognized as a communication vehicle for the entire project team, subject matter experts (SMEs), and stakeholders who are working on testing activities, or those who need to be aware of the information it contains. This document and its content provide guidance for Testers at every level of testing. It should be used in tandem with all other project management procedures.

## Test Objectives

This MTP supports the following objectives:

* To provide test coverage for 100% of the documented requirements
* To execute 100% of the test cases during all software quality assurance (SQA) testing efforts
* To create, maintain, and control the test environment within the Development, Pre-

Production and Production environments

* To execute Section 508 Testing
* To include End User Testing (EUT) to test the effectiveness of the system for use by VA Mental Health SMEs and Suicide Prevention Coordinators (SPCs)
* Identify and report any potential risks that might impede testing
* Provides a test methodology that works within the Continuous Integration (CI) process and ensures all automated tests scripts are consistently passing
* Provides the ability to trace test scripts back to requirement changes using a Test Driven Development (TDD) approach

# Testing Approach

Testing strives to discover defects in the system or software by exercising the code. Testing activities are distributed throughout the development, implementation, and enhancement of the application.

The Test Team will conduct manual testing and automated testing. Manual testing will involve creating detailed Test Scripts which the Testers will execute. These Test Scripts will be written in clear and literal language to provide a non-technical approach to testing.

Automated testing will be done utilizing open source solutions, such as Cucumber and Jasmine, where appropriate. These testing solutions give Test Team members flexibility and are tools commonly used on other VACI innovation projects.

The goal of the automated Test Scripts is to provide as much test coverage using automation as appropriate. Depending on the system complexity and environment constraints, manual testing will be used to fill in the gaps where automated testing is not feasible or does not provide adequate coverage in ad-hoc or unique scenarios found in the application.

## TDD

The project will utilize a TDD process in alignment with an Agile methodology. The TDD will focus on the creation of unit tests by the development team for each new function. The development team will then produce functionality that is confirmed by the previously created unit test through a CI process.

### Processes

**Test preparation**

* Validate the requirements
* Validate the User Stories
* Build Test Suites, Test Cases, and supporting Test Scripts
* Provide traceability from each Test Case to User Stories and requirement(s)

**Build process**

* Product Build – the software is built on the VA Cloud web server environment using Jenkins, Node, Grunt, and IISNode
* The Test Scripts will be built in Cucumber and executed against the product using Jenkins

**Testing**

* New Development
* Multiple levels of testing are performed: Software Engineers are responsible for conducting Unit Testing for their code before it is checked in and deployed. After the code is checked in, a build is completed and deployed to the Development environment in the VA Innovation Sandbox
* Builds that successfully pass Unit Testing will then be functionally tested utilizing automated test scripts and formal test cases validating User Stories and associated requirements.

**Defect Management**

* A defect is defined as any unwanted behavior by the application or when a deviation from documented requirements is observed
* The development team will use Jira as the Defect Management or tracking tool
* Defects identified within the course of a Sprint can be added to the current Sprint or reviewed and planned for a future Sprint
* When a Developer confirms that their code is complete for a feature and the code is promoted to the Test environment through a build and deployment, the Testers can begin formal testing on the code. Actual results shall be documented when the test result of a Test Case deviates from the expected result.
* Defects are identified by the Testers during test execution. Defects can be identified when the following conditions occur:
  + Actual results do not correspond to expected results described in the Test Cases
  + Processing events do not correspond to what is described in the User Stories and the requirements
  + Other defects are found, such as field mapping and validation, incorrect error conditions, and usability issues
  + The Test Case is unable to be repeated
* Test results are recorded and results that are not expected may identify issues and defects. Before determining if a test result is indeed a defect, the results must be verified with the Developers and stakeholders. Most often a demonstration executing the Test Scripts to show the unexpected result is needed.
* Defects are verified against the requirements to determine if there is a missed requirement or an out of scope requirement, or that the defect is indeed valid. The Test Cases are then reviewed to determine if the test is valid.
* In the case when a defect arises due to a missing requirement, the Test Lead will review the test results with the Developer and stakeholders to determine if additional requirements are needed or amendments to current requirements are necessary.
* For valid issues, the Developer may modify the code and attempt to fix the issue before the end of the Sprint. If the issue is fixed before the end of the Sprint and the Test Lead verified the fix during the end of Sprint testing, then the issue is resolved with no defect being entered.

Table 1: Defect Priority Levels

|  |  |
| --- | --- |
| **Priority Level** | **Description** |
| Blocker | Critical functionality not implemented correctly; user cannot meet the business objectives; a work-around (temporary fix) is ***NOT*** available |
| Critical | Errors found which result in the user not being able to meet the business objectives; a work-around (temporary fix) is available |
| Major | This is a moderate defect, which can cause some deviations from the requirements; deviation is not serious and does not require a work around |
| Minor | Minor defect, which causes the user some inconvenience but does not alter functionality (e.g. spelling error, user interface [UI] issue, aesthetics, or other minor issues) |
| Trivial | Cosmetic problem like misspelled words or misaligned text |

* When logging a defect in the Product Backlog, the following defect information should be captured in the User Story or task description:

Table 2: Defect Attributes

|  |  |
| --- | --- |
| **Priority** | The priority corresponds to how the defect will affect the system overall; its overall impact on the whole system if not fixed. |
| **Summary Details** | Include those which are applicable of the following:   1. Which Epic or User Story and requirement the defect is entered against 2. The Internet browser being used; display type/settings, and values 3. The environment where testing is being performed 4. Build number 5. User name and password of the test account used; note the roles and permissions for the user 6. Date and time (an approximate estimation will suffice) so the logs can be checked 7. The “requirement” or User Story that will link the defect for traceability 8. What are the steps to recreate the defect 9. Any screen shots and also data, if it is back end, and test data (user credentials/file numbers); list the titles of the documents linked to the defect 10. Any queries used, name of the database validating against, and the table/field names 11. The actual results obtained which revealed the defect and why those results show a flaw, failure, or fault in the software |

# Items to Be Tested

* IRDS Dashboard
* Direct Messaging
* Reach Database
* Analytics Platform

## Overview of Test Inclusions

The table below includes an overview of the major user requirements, or “Epics,” associated with the proposed solution. A full listing of the project’s Product Backlog is maintained in Jira. In addition, a snapshot of the project’s major functional requirements will be included in the project’s RTM.

Table 3: Functional Requirements

| JIRA Key | **Description** |
| --- | --- |
| [PR-158](https://opensourceehr.atlassian.net/browse/PR-158) | As an Outreach Provider, I want to be sent secure notifications via a Direct Message of at-risk Veterans and populations so I can provide outreach services to these groups. |
| [PR-346](https://opensourceehr.atlassian.net/browse/PR-346) | As an Outreach Provider, I want to view a Direct Message that highlights a Veteran at high risk for suicide so I can provide outreach services to them. |
| PR-160 | As a member of VA leadership, VA Center of Excellence for Suicide Prevention staff, VA Mental Health leaders, and VA Suicide Prevention Coordinators (hereby indicated as a "Dashboard User") I want to view a surveillance Dashboard with results produced from the continuous monitoring and processing of linked data sources so I can monitor and understand Suicide Outreach outcomes. |
| PR-1537 | As a Dashboard User, I want to view a Veteran's recent medical diagnoses from within the Dashboard so I can view the Veteran's recent issues at a glance and make outreach and care decisions. |
| [PR-1571](https://opensourceehr.atlassian.net/browse/PR-1571) | As a Dashboard User, I want widgets to be enabled to communicate with each other across the dashboard. (e.g. Veteran Roster to Supporting Widgets like Appointments or Medications). |
| [PR-1789](https://opensourceehr.atlassian.net/browse/PR-1789) | As a system administrator, I want the IRDS to generate reports/logs when new data is imported into the system. |
| [PR-1267](https://opensourceehr.atlassian.net/browse/PR-1267) | As a Dashboard user, I want to see a Veteran's contact information so that I can contact them to provide outreach and intervention services. |
| [PR-1268](https://opensourceehr.atlassian.net/browse/PR-1268) | As a Dashboard user, I want to see a Veteran's emergency contact information so that I can contact them to provide outreach and intervention services. |
| [PR-1380](https://opensourceehr.atlassian.net/browse/PR-1380) | As a Dashboard user, I want the ability to update a Veteran's outreach status so that I can track what Veterans have not yet been contacted, which Veterans are in outreach / intervention services, which Veterans have refused service, etc. |
| [PR-1405](https://opensourceehr.atlassian.net/browse/PR-1405) | As a Dashboard User, I want to view a Veteran's medications from within the Dashboard so I can view the Veteran's prescriptions at a glance and make outreach and care decisions. |
| [PR-1407](https://opensourceehr.atlassian.net/browse/PR-1407) | As a Dashboard User, I want to view a Veteran's medical appointments from within the Dashboard so I can view the Veteran's appointments at a glance and make outreach and care decisions. |
| [PR-1248](https://opensourceehr.atlassian.net/browse/PR-1248) | As a Dashboard user, I want to view the change in a Veteran's pain level over time so that I can help them manage pain, if needed. See attachment in Jira for a screen capture from a pain management app. |
| [PR-356](https://opensourceehr.atlassian.net/browse/PR-356) | As a Dashboard User, I want to pick and choose which screen elements I see on the Dashboard, so I can first see only the data that is important to me. |
| [PR-357](https://opensourceehr.atlassian.net/browse/PR-357) | As a Dashboard User, I want to move screen elements I see on the Dashboard so I can customize the look of the Dashboard to suit my preferences. |
| [PR-881](https://opensourceehr.atlassian.net/browse/PR-881) | As a Dashboard User, I want to see a line graph that shows the change in how many Veterans are at the top .1%,or 5% of the risk stratification model over time. |
| [PR-877](https://opensourceehr.atlassian.net/browse/PR-877) | As a Dashboard User, I want to see a widget that allows me to view a menu showing how many Veterans are within a certain risk stratification |
| [PR-876](https://opensourceehr.atlassian.net/browse/PR-876) | As a Dashboard User, I want to see a widget that allows me to view a "roster" (or list) of Veterans that have been identified by the application as high risk. |
| [PR-996](https://opensourceehr.atlassian.net/browse/PR-996) | As a Dashboard User, I want to be presented "Clinical Decision Support" information related to a Veteran's specific information. |
| [PR-994](https://opensourceehr.atlassian.net/browse/PR-994) | As a Dashboard User, I want to view individual Veteran information that is relevant to suicide outreach, intervention, and care, so I can make clinical care decisions for treatment of the Veteran. |
| [PR-349](https://opensourceehr.atlassian.net/browse/PR-349) | As a Dashboard User, I want to log in to the Perceptive Reach application. |
| [PR-1676](https://opensourceehr.atlassian.net/browse/PR-1676) | As a Dashboard user, I want to enter my credentials (username and password) to log in to the Dashboard |
| [PR-1887](https://opensourceehr.atlassian.net/browse/PR-1887) | As a Dashboard user, I want to be presented with a "landing page" where I can log in to the application. |
| PR-1788 | As an IRDS dashboard user I expect the system will automatically terminate my session after a specified period of inactivity (ex. 10 minutes) such that an unauthorized user could not use my system if I walk away from my computer. |
| [PR-1677](https://opensourceehr.atlassian.net/browse/PR-1677) | As a Dashboard user, I want my log in to be rejected if my credentials are not accurate or valid. |
| [PR-1678](https://opensourceehr.atlassian.net/browse/PR-1678) | As a Dashboard user, I want log out of the Dashboard to end my session. |
| [PR-1679](https://opensourceehr.atlassian.net/browse/PR-1679) | As a system administrator, I want to grant users default access to the Dashboard (Individual Veteran and Facility Level Views). |
| [PR-1680](https://opensourceehr.atlassian.net/browse/PR-1680) | As a system administrator, I want to grant users supervisory access to the Dashboard (Surveillance plus all other views). |
| [PR-1681](https://opensourceehr.atlassian.net/browse/PR-1681) | As a system administrator, I want to change a user's access level if appropriate. |
| [PR-1705](https://opensourceehr.atlassian.net/browse/PR-1705) | As a Dashboard user, I want to be logged in to the Dashboard via VA Single Sign On automatically |
| PR-1569 | As a Dashboard User, I want to be able to save my Dashboard settings / customizations when I log out of the application and see them again when I have logged back |
| [PR-351](https://opensourceehr.atlassian.net/browse/PR-351) | As a Dashboard User, I want to see data from my "home" facility when I log in. |
| [PR-161](https://opensourceehr.atlassian.net/browse/PR-161) | As a Researcher, I want to access research tools and data in the application so I can perform research-related tasks and projects. |
| [PR-724](https://opensourceehr.atlassian.net/browse/PR-724) | As a Researcher, I want to access the data in the underlying database using BIRT. |
| [PR-725](https://opensourceehr.atlassian.net/browse/PR-725) | As a Researcher, I want to access the data in the underlying database using R. |
| [PR-726](https://opensourceehr.atlassian.net/browse/PR-726) | As a Researcher, I want to access the data in the underlying database using MySQL. |
| [PR-162](https://opensourceehr.atlassian.net/browse/PR-162) | As a Reporter, I want to generate reports using the data and automated tools in the application so I can use reports as management and communication tools. |
| [PR-889](https://opensourceehr.atlassian.net/browse/PR-889) | As a Reporter, I want to generate reports from the data in the underlying database using BIRT. |
| [PR-890](https://opensourceehr.atlassian.net/browse/PR-889) | As a Reporter, I want to generate reports from the data in the underlying database using R. |
| [PR-891](https://opensourceehr.atlassian.net/browse/PR-889) | As a Reporter, I want to generate reports from the data in the underlying database using MySQL. |
| [PR-892](https://opensourceehr.atlassian.net/browse/PR-889) | As a Reporter, I want to access a set of "canned" reports that are automatically generated on a regular basis. |
| [PR-163](https://opensourceehr.atlassian.net/browse/PR-163) | As a User, I want to edit, add to, and create new IRDS Risk Stratification Models and mapping to interfaces so the application can be updated over time. |
| [PR-1876](https://opensourceehr.atlassian.net/browse/PR-1876) | As a Dashboard user, I want the ability to notify system administrators if a Veteran should be assigned to a different VAMC. |
| [PR-1877](https://opensourceehr.atlassian.net/browse/PR-1877) | As a Dashboard user, I want to send a notification to Perceptive Reach administrators if I think a Veteran should be reassigned to a different VAMC. |
| [PR-1879](https://opensourceehr.atlassian.net/browse/PR-1879) | As a system administrator, I want to transfer a Veteran's assignment from one VAMC to another based on the requests of Dashboard users. |
| [PR-1880](https://opensourceehr.atlassian.net/browse/PR-1880) | As a Dashboard user, I want to receive a Virtual Lifetime Electronic Record (VLER) Direct Message notifying me when a Veteran has been assigned to my VAMC in the application. |
| [PR-1881](https://opensourceehr.atlassian.net/browse/PR-1881) | As a system administrator, I want to view a log of Veteran VAMC change requests and transfers so I can review and audit the log as needed. |
| [PR-1787](https://opensourceehr.atlassian.net/browse/PR-1787) | The IRDS system shall have error detection and handling mechanisms. |
| [PR-1870](https://opensourceehr.atlassian.net/browse/PR-1870) | As a System Administrator, I want the application to detect and notify me of network errors so I can troubleshoot and resolve the errors in an expeditious manner. |
| [PR-1871](https://opensourceehr.atlassian.net/browse/PR-1871) | As a System Administrator, I want the application to detect and notify me of CDW interface errors so I can troubleshoot and resolve the errors in an expeditious manner. |
| [PR-1872](https://opensourceehr.atlassian.net/browse/PR-1872) | As a System Administrator, I want the application to detect and notify me of remote system errors so I can troubleshoot and resolve the errors in an expeditious manner. |
| [PR-1873](https://opensourceehr.atlassian.net/browse/PR-1873) | As a System Administrator, I want the application to detect and notify me of data errors (rx data corruption or erroneous) so I can troubleshoot and resolve the errors in an expeditious manner. |
| [PR-353](https://opensourceehr.atlassian.net/browse/PR-353) | As a Dashboard User, I want to access Dashboard "views" that summarize data for a specific facility service area, or higher level view. |
| [PR-1684](https://opensourceehr.atlassian.net/browse/PR-1684) | As a supervisor, I want to see a summary of how many Veterans are in the top .1% and 5% risk stratifications. |
| [PR-1685](https://opensourceehr.atlassian.net/browse/PR-1685) | As a supervisor, I want to see a summary of how many Veterans in the top risk stratifications are associated with each VA Medical Center. |
| [PR-1686](https://opensourceehr.atlassian.net/browse/PR-1686) | As a supervisor, I want to see a summary of how many Veterans are in each age group in the top .1% and 5% risk stratifications. |
| [PR-1687](https://opensourceehr.atlassian.net/browse/PR-1687) | As a supervisor, I want to see a summary of how many Veterans in the top .1% and 5% risk served in the various US Military Branches. |
| [PR-1688](https://opensourceehr.atlassian.net/browse/PR-1688) | As a supervisor, I want to see a summary of how many Veterans are every outreach and intervention status in the top .1% and 5% risk stratifications. |
| [PR-1690](https://opensourceehr.atlassian.net/browse/PR-1690) | As a supervisor, I want to see a summary of how many Veterans in the top .1% and 5% risk stratifications have specific diagnoses. |
| [PR-1692](https://opensourceehr.atlassian.net/browse/PR-1692) | As a supervisor, I want to see a summary of how many Veterans in the top .1% and 5% have been flagged as high risk for suicide in SPAN. |
| PR-2079 | As any IRDS user, I want to view the standard VA 6500 Handbook warning before I log in to the system. |
| PR - 2080 | As any IRDS user, I want to click a checkbox when I log in, indicating that I agree to the terms and conditions described in the system security warning on the login page. |
| PR - 2081 | As a system administrator, I want any user who attempts three invalid logins to be locked out of the system. |
| PR - 2082 | As a system administrator, I want the application to reject any login attempts by non-system administrator users who already have an active session open. |
| PR - 2083 | As a system administrator, I want the capability to have up to 3 active instances of the application running at once. |
| PR - 2084 | As any IRDS user, I want to be automatically logged out of the application after 15 minutes of inactivity. |
| PR - 2085 | As a system administrator, I want to view a log of past user activities including user changes to the Dashboard, configurations, logins, etc. |
| PR - 2086 | As a system administrator, I want the application to alert me if the activity log database is nearing capacity so I will know when older data will soon be overwritten. |
| PR - 2087 | As a system administrator, I want the application to fully "close" when / if the application fails, so that no users can view, modify, or interact with the application when it has failed. |
| PR -2088 | As a system administrator, I want the application to reject user login attempts that have the attributes listed in the acceptance criteria of this story. |
| PR-3437 | As a clinical care team member, I want to see a widget that allows me to enter and save new patient information in the IRDS Dashboard. |
| PR-3721 | A surveillance level widget that shows the upper and lower predicted bounds for suicide attempts at a VA facility. |
| PR-3861 | As a VISN level (or other higher level user), I want the ability to view data from multiple specific facilities so I can make program decisions across those facilities. |
| PR-3860 | As a Dashboard user, I want the option to print the contents of a roster and / or widget from the Dashboard. |
| PR-3965 | As a suicide prevention leader, I would like to see a single view that contains mortality information in a graphical form so that I can visually assess trends and patterns and statistics related to suicide. |
| PR-4041 | As a Dashboard user, I want patients to be accurately associated with my facility so I can reach out to them in my service area. |
| PR-4302 | As an IRDS system administrator, I want the application to be 6500 compliant. |
| PR-4411 | As a Dashboard user, I want patients to stay on the Dashboard after they have been identified. |
| PR-4166 | As a Dashboard user, I do not want Veterans to be automatically removed from the Dashboard as a result of a risk score calculation. As a Dashboard user, I want to be the only one that has the ability to remove Veterans from the Dashboard. |
| PR-4505 | As an IRDS Dashboard user, I would like to be able to see community resources on the Individual view so I can recommend them to a Veteran. |
| PR-4599 | As a Perceptive Reach user, I want the current outreach status drop-down menu options to be updated so they better align with my workflow. |

## Overview of Test Exclusions

Any testing not described within the PWS or testing that requires 3rd party accreditation will be excluded from this document.

## Product Component Test

Each Developer is responsible for fully testing all code that is delivered against a User Story. Developers must test each component to ensure that proper error handling and fail-over conditions are covered.

## Component Integration Test

The Developers responsible for the Perceptive Reach project must test each component to ensure that proper error handling and fail-over conditions are covered within the current provider workflow.

## System Tests

Unit, integration and functional tests are used for system testing. The system is validated at unit, integration, and functional level. Testers will work with development and analysts teams to understand the acceptance criteria to create manual and automated test scripts.

# Automated Testing

Automated testing in the IRDS application will be created utilizing Cucumber scripts which use Ruby for processing and executing. Cucumber scripts are written with user behaviors in mind by focusing on elements of Human Centered Design (HCD). The scripts are composed and executed using Jenkins to automate the execution, recording of result, and reporting all in one platform. For reference, an example of an automated Cucumber automated testing file is listed below.

## Regression Testing

Regression testing for the IRDS application is conducted to verify that newly deployed functionality and bug fixes are working as expected, and have not affected previously existing functionality. The regression test plan consists of the automated test suite, and manual tests that were executed to cover functionality that cannot be automated. The Perceptive Reach team first focused on testing bug fixes, then verifying new functionality, and finally testing a sample of functional test cases from prior releases. Please see the Regression Test Plan for more information.

## Feature File Example

The following example can be executed manually or automated through the use of Cucumber and testing tools.

Scenario: As an End-User I open a web browser and Log into the Perceptive Reach Dashboard <Include Requirement Number>

Given I navigate to the http://<application-url>/

Then I should see "Perceptive Reach Login"

Then I put in "email" field as

Then I put in "password" field as

And I click on "Login" button

Then I should see "National View"

And I click on "Logout" button

## End User Testing

End User Testing consists of real world scenarios relevant to the end-users. These real world scenarios are based on various requirements and ensure proper coverage of user operations during testing. Test cases are detailed examples of how the system is supposed to function when the requirement is complete. Testing is most effective if it is carried out in an environment that closely resembles the user’s environment or the Production environment. EUT is intended to focus on the functionality and usability of the application, rather than just the technical aspects.

Testing will focus on the end-user experience through browser emulation and user interaction. End User Testing will take place over the internet to simulate bandwidth restrictions users many experience in production. The performance metrics will be gathered through manual testing at the browser and automated testing using Cucumber and additional testing tools.

The test team will work with the VA PM, SMEs, SPCs, and end-users while leveraging existing artifacts such as User Stories, requirements, and test cases to develop a comprehensive scenario that directly relates to end-user activities.

## Pilot Testing

Pilot Testing will focus on the usage of live data and the process by which it is imported and scored through the analytical model implementation in the IRDS Database. The imported data will be manually compared between the IRDS and CDW data sets to confirm accuracy. The IRDS scoring computed through the SQL SISS packages will be validated using R to compare results. Once the data import and scoring has been verified, the next phase of Pilot Testing will focus on the dashboard application. Pilot Testing will use manual and automated tests to confirm functionality and usability of the PR Dashboard.

IT Pilot testing includes a series of testing procedures that allow developers to test aspects of the system and ensure correct functionality. This testing process has been organized in a way that fosters objectivity as no developers are testing personally authored code or functionalities. This process has been separated into individual test plans created to support each IT Pilot testing initiative listed below. A schedule has been created listing the IT Pilot testing that will take place “Pre” and “Post” ATO. For purposes of efficiency, some tests will be conducted both before and after the ATO is awarded. Aspects of IT Pilot testing including test creation, test execution, and test review have been recorded in Jira and assigned to an individual responsible for completing that specific task. Following the execution of the test plans below, a test result review will take place and the findings will be recorded in an IT Pilot result report to be delivered at a later date.

**Regression Test Plan**

This test plan is meant to test the IRDS system to ensure that all bug fixes and newly deployed functionality are functioning as expected, and are not negatively impacting currently existing functionality. This test suite is updated and executed prior to each deployment, with the goal of providing full test coverage of all components of the application.

**6500 Requirements Test Plan**

This test plan is meant to test the IRDS system to make sure it meets 6500 requirements and acts correctly in regards to log-in and log-out situations. This test will make sure that the log-in page is formatted properly and displays all required information. It also verifies that the log-in page behaves correctly when the user provides incorrect information or actions. The items that will be tested are listed below:

* Verify that any IRDS User is able to view the standard VA 6500 Handbook before logging into the system.
* Verify that a User is logged out of the system if they have three invalid logins.
* Verify that a standard user can open and maintain no more than one active session at any given time.
* Verify that a system administrator can open and maintain up to three concurrent and active instances of the IRDS application.
* Verify that an IRDS user is automatically logged out of the application after 15 minutes of inactivity.
* Verify that the application rejects user login attempts that exceed a name character limit of 32 characters.

Verify that users from different domains can successfully login to the application.

Verify that a standard user's status is updated to 'Active' by the system administrator after three unsuccessful login attempts.

WinCollect log validation and testing was performed in the option year in coordination with 6500 testing to ensure all necessary user actions are being logged on the server and the application is VA Handbook 6500 compliant. WinCollect logs were requested from AITC for the day 6500 testing was executed.

* Validate in the WinCollect logs that a record appears displaying the connection to database server via Remote Desktop (RDP).
* Validate in the WinCollect logs that a record appears displaying the connection to database server thru' a SQL Connection using SSMS.
* Validate in the WinCollect logs that a record appears indicating that information was exported from the Patient Roster by VAMC widget.
* Validate in the WinCollect logs that a record appears indicating that a System Administrator modified or added Perceptive Reach users in the Database.
* Validate in the WinCollect logs that a record appears verifying that SSIS package(s) were run.
* Validate in the WinCollect logs that a record appears verifying that storage procedure(s) were run.

**Attempt Prediction Chart Widget Test Plan**

The purpose of this test plan is to validate functionality related to the Attempt Prediction Chart widget available to supervisors on the IRDS Dashboard. This widget uses data from the Suicide Prevention Applications Network (SPAN) and a predictive algorithm to display the number of suicide attempts at a facility over time, in addition to the expected upper and lower bounds of the number of suicide attempts at a facility over time. The widget displays these three data points on a line graph. Users can observe if the number of suicide attempts at a facility has spiked or valleyed beyond the expected bounds, and use that information to investigate to see if there is something happening at the facility that could justify a correction or intervention. The items that will be manually tested are listed below.

* Verify that the Attempt Prediction Chart widget is located on the Surveillance view and is available to all supervisors.
* Verify that the widget displays a line chart with an X and Y axis after a facility is selected in the Facility Roster widget on the Surveillance view.
* Verify three black lines appear across the chart indicating the upper and lower bounds of the attempt prediction, with the prediction line in the middle.
* Verify a blue line appears with blue and red points defining the Upper Limit, Lower Limit, Best Fit and Actual values displayed when we hover on the points.

**Break Test Plan**

The goal of this test plan is to examine how the IRDS system responds to unplanned situations where aspects of the database or network malfunction. This test will expose areas of improvement for the IRDS application and allow preventative measures to be taken to enhance future performance when encountering similar situations where the system is put under stress. The items that will be tested are listed below:

* Test IRDS application ability to recover from a CDW data push connection error.
* Test IRDS application ability to recover from a SDR data pull connection error.
* Test IRDS application ability to recover from an import of corrupt data.
* Test IRDS application ability to recover from a VLER direct shutdown.
* Test IRDS application ability to recover from a Database shutdown.
* Test IRDS application ability to recover from an unexpected server restart.
* Test IRDS application ability to recover from a database failure in the Perceptive Reach dashboard.
* Test IRDS application ability to recover from a situation where HealthIndicators.gov is not accessible.

**Data Entry Widget Test Plan**

The purpose of Data Entry widget testing is to validate functionality related to the Data Entry widget on the IRDS Dashboard. This widget introduces an informal reporting component to the IRDS Dashboard allowing users to enter and save patient-related information in the IRDS Database. The Data Entry widget displays information entered by the user and official information from the systems of record. The items that will be manually tested are listed below.

* Verify that the Data Entry widget is located on the Individual view and is available to all users.
* Verify that the Data Entry widget has the following sections, each of which should include a ‘User Notes’ field that can be edited by the user.
  + (1) High Risk Flag Information
  + (2) Mental Health Provider Information
  + (3) Safety Plan Information
  + (4) General Notes
* Verify that for all of the sections listed above, the user is able to edit the ‘User Notes’ fields and save their entered data by selecting the ‘Add Data’ button.
* Verify that for all of the sections listed above, the user is able to select the left arrow to review the previous information entered and the right arrow to see the next information entered.
* Verify that the following read-only system of record fields are available for view
  + (1) ‘SPAN Records’ (directly under ‘High Risk Flag Information’ field)
  + (2) ‘VistA Records’ (directly under ‘Mental Health Provider Information’ field)
  + (3) ‘VistA Records’ (directly under ‘Safety Plan Information’ field)
* Verify that for all of the sections listed above (except the ‘VistA Records’ field associated with the ‘Mental Health Provider Information’ section), the user is able to select the left arrow to review historical system of record information and the right arrow to see more recent historical system of record information.

**Database Test Plan**

The purpose of Database Testing is to validate functionality related to data import functionality as outlined in the Perceptive Reach PWS and describe the testing approaches being used. This test plan is meant to test multiple aspects of the IRDS database in the pre-production environment. The tester will perform multiple tests that verify the IRDS database is functioning accordingly. This plan will test the IRDS’s capability to successfully receive information from pre-existing databases like CDW and SDR. Database Testing includes a series of testing procedures that allow developers to test aspects of the server and ensure correct functionality. All IRDS database testing will be performed in the Reach pre-production database on the database server. Following the execution of the test plans below, a test result review will take place and the findings will be recorded in an IT Pilot result report to be delivered at a later date. The items that will be tested are listed below:

* Validate the backup functionality of the IRDS MS SQL Server and Verify that the backup of the database is processed and saved accurately.
* Verify and validate the restoration functionality of the IRDS MS SQL Server and a message is displayed on a successful restoration.
* Verify the ability to create a test table to be used to store data pushed from CDW to IRDS.
* Verify that you can execute a test ETL package in the CDW workspace and that the package runs successfully without any errors.
* Verify the results of the test ETL package in the CDW workspace and validate that the data displayed in the databases match.
* Verify that you can create a test table to be used to store data pulled from SDR to IRDS.
* Verify that you can execute a test ETL package on the IRDS Database Server related to SDR and that the package runs successfully without any errors.
* Verify the results of the test ETL package on the IRDS Database Server related to SDR and validate that the data displayed on the databases match.
* Verify that the system can handle corrupt data testing.
* Verify that a test ETL package attempts to force corrupt data into the IRDS Database Server the package will not complete with corrupt data present and will provide a list of error messages identifying the corrupt data issue.
* Validate the handling of corrupt data by the IRDS MS SQL Server and verify that a description of the error is displayed.
* Validate the configuration of the IRDS Database Server base Operating System.
* Validate data in the IRDS system and verify that correct information is displayed.

Database testing was also carried out in the Perceptive Reach option year. The base year database test plan was updated in order to be applicable for the current IRDS system. All bullets listed above were tested in the option year including the items listed below. Any defects identified during testing were announced to the team and addressed.

* Validate that data in the Data Entry Widget in the IRDS Dashboard displays correct information.
* Validate that historical entry functionality in the database related to the Data Entry Widget is working correctly.

**End to End Scenario Test Plan**

The goal of the End to End Test Plan is to validate an “end to end” type use case for the Perceptive Reach application and its components – the IRDS, VLER Direct and Upstream Notification functionality, Surveillance Dashboard, Reporting, and Research. The items that will be tested are listed below:

* Verify VLER and Upstream:
  + Verify that you can login to VLER Direct.
  + Verify that a message appears in your inbox indicating the number of Veterans, those Veterans have a status of “NEW”, and that Veterans are labeled as either “MIDDLE” or “TOP” risk.
  + Verify that the message includes a clickable link that directs the user to the Dashboard when clicked.
  + Verify that the user is prompted to login (if required).
  + Verify that the Dashboard loads the user’s default dashboard view automatically.
* Verify Surveillance: Verify after successful user login, the widgets on the Perceptive Reach Individual view are present and displaying actionable information
  + Login to the Perceptive Reach application and verify that the Dashboard page is displayed.
  + Verify that the view ribbon displays.
  + Verify that you can access the “Patient Roster by VAMC” button after clicking on the “Individual View” tab in the ribbon.
  + Verify the Veteran Contact widget is displayed within the webpage that includes all necessary Veteran emergency contact information (address, phone number, alternate phone number).
  + Verify that the “Recent Medications” Widget is displayed within the webpage.
  + Verify that the “Recent Diagnoses” Widget is displayed within the webpage.
  + Verify that the “High Risk Flags” Widget is displayed within the webpage.
  + Verify that the “Recent Appointments” Widget is displayed within the webpage.
  + Verify that the “Clinical Decision Support (CDS)” is displayed within the webpage.
* Verify Report: Verify that the R, BIRT, and MySQL tool applications provide the option of exporting information into numerous data files.
* Verify Research: Verify that the system is equipped with MySQL, R, and BIRT applications on the Remote Desktop in order for users to find information on a certain topic.

**End User Test Plan**

The goal of the following test plan is to validate a high level “end to end”-type use case for a typical clinical end user (such as a Suicide Prevention Coordinator) of the Perceptive Reach application. The primary application components included in this test plan are VLER Direct + Upstream Notification functionality and the Surveillance Dashboard. The items that will be tested are listed below:

* Verify that a user can successfully log-in to VLER direct and check a message in their Inbox.
* Verify that the message subject line reflects the number of Veterans.
* Verify that Veterans have a status of NEW and are labelled as a TOP risk.
* Verify that clicking the Dashboard link opens the Dashboard page.
* Verify that the user can access the Individual View.
* Verify that the patient can view the Patient Roster by VAMC.
* Verify that the Patient Contact widget displays within the webpage.
* Verify that the Emergency Contact widget is displayed within the webpage.
* Verify that the system logs the user out when they press the log-out button.

**Health Indicators Test Plan**

The purpose of the health indicators test plan is to validate functionality related to Health Indicators data import functionality. The Health Indicator Widget will display a table of information correlating the number of suicide deaths per 100,000 people for the applicable year and will be grouped by Age Range, Gender, and Ethnicity. The items that will be tested are listed below

* Verify that the data displayed on the Health Indicators Widget matches the data displayed on the Health Indicators site for the “Suicide Deaths per 100,000” table for the year “2013”.
* Verify that the Health Indicators Widget exists and includes a data table of 4 columns:
  + (1) Age Range
  + (2) Gender
  + (3) 2013 Total Suicide Deaths per 100,000
  + (4) Ethnicity
* Verify that the Health Indicators Widget provides the URL to source of data on HealthIndicators.gov website.

**Reporting Scenario Test Plan**

The purpose of this test plan is to provide an overview of test scenarios and execution steps for users to create and export reports using MySQL, BIRT, and R utilizing the IRDS database. The items that will be tested are listed below:

* Verify that the system is capable of understanding SQL queries on the Remote Desktop in order for users to find information on a certain topic.
* Verify that the R tool application provides the option of exporting information into numerous data files (i.e. xlsx, XML).
* Verify that the BIRT application allows the user to export various report files to various formats (DOC, DOCX, HTML, PDF, PPT, XLS, etc.).

**Research Scenario Test Plan**

This plan is an overview of test scenarios and execution steps for the Research platform. The Research platform allows a user to perform research on the available data in the IRDS application. The items that will be tested are listed below:

* Verify that the system is capable of understanding SQL queries on the Remote Desktop in order for users to find information on a certain topic.
* Verify that the R tool application provides the option of exporting information into numerous data files (i.e. xlsx, XML).
* Verify that the BIRT application allows the user to export various report files to various formats (DOC, DOCX, HTML, PDF, PPT, XLS, etc.).

**Section 508 Test Plan**

The purpose of this 508 section Test Plan is to validate functionality related to accessibility features that allow users to access the application as outlined in the Perceptive Reach PWS and describe the testing approaches being used. The goal of preliminary testing is to identify and fix section 508 compliance issues within the development phase to minimize the amount of subsequent rework and identify problem areas that would be raised by the VA’s Section 508 Certification Group. The items that will be tested are listed below:

* Verify that all screens/fields are readable by a screen reader, for unsighted users.
* Verify that all screens are keyboard accessible (users can complete the screens without the use of a mouse.
* Verify that the tab order is logical and that the unsighted user can fully use the application in the same manner (with full functionality) as a sighted user.

**Surveillance Scenario Test Plan**

The purpose of this test plan is to provide an overview of test scenarios and execution steps for the surveillance user interaction model. This test plan will ensure that all available dashboard views (based on access level) are accessible from the main Dashboard webpage. The items that will be tested are listed below:

* Verify that the login screen has an option to enter a username and password with a terms and conditions text box and ability to successfully login to the application with valid credentials.
* Verify that after successful user login, the default widgets on the Perceptive Reach Facility Level Dashboard View are present and displaying the correct information.
* Verify that after successful user login, the default widgets on the Perceptive Reach Individual Level Dashboard View are present and displaying the correct information.
* Verify the ability to successfully logout of the application and that the user is returned to the main login screen.

**Upstream At-Risk Notification Test Plan**

The goal of this test plan is to validate that processes put in place to notify health professionals about at-risk Veterans are functioning properly. The upstream notification system within the IRDS application automatically notifies health professionals about at-risk Veterans, allowing them to provide appropriate services to those Veterans in a timely manner. This test is used to verify that the automated messaging system required in this process is operating correctly and that facets related to this messaging system are also working properly. The items that will be tested are listed below:

* Verify that the application automatically notifies VA health professionals of newly identified Veterans through a direct message (DM) sent to their VLER direct email accounts.
* Verify that the application places Veterans in the TOP and MIDDLE risk groups when appropriate and records these statuses in the DM sent to health professionals’ VLER direct accounts.
* Verify that the application shows which Veterans moved from the MIDDLE to TOP risk group.
* Verify that the VLER DM includes all necessary Veteran identification information.
* Verify that a clickable link exists in the VLER DM that directs the user to the Dashboard.

**User Role Test Plan**

The goal of this test plan is to verify that all personnel using the system have full access to functionalities related to their specific roles (i.e. Supervisor, Clinical Care Supervisor (CCS), Clinical Care Team Member (CCTM), Reporter, Researcher, and System Administrator). Access to these applications is important and allows users to fulfill the duties associated with their individual roles. The items that will be tested are listed below:

* Verify the Supervisor role can access the Facility, Surveillance, and Individual Level views only and validate that the widgets associated with each of those views are displayed.
* Verify the CCS role can access multiple Facilities and the Individual Level view and validate that the widgets associated with each of those views are displayed.
* Verify that the CCTM role can access the Facility and Individual Level views only and validate that the widgets associated with each of those views are displayed.
* Verify the Reporter role can access the Facility, Surveillance, and Individual Level views only and validate that the widgets associated with each of those views are displayed.
* Verify that the Researcher role can access the R / BIRT / MySQL Database Access only and validate that the widgets associate with each of those views are displayed
* Verify the System Administrator role can access all dashboard views and validate that they can view all widgets displayed for each specific view

**VLER Test Plan**

The purpose of this test plan is to provide an overview of scenarios and execution steps to review VLER’s underlying logic and processes to correctly deliver messages to users. Verifying the correct functionality of this component is critical as this is the main way in which health professionals will be notified of at-risk Veterans. The items that will be tested are listed below:

* Verify that the model runs and identifies the number of TOP risk Veterans that have not been previously identified
* Verify that the model runs and identifies the number of MIDDLE risk Veterans that have not been previously identified
* Verify that the model identifies those Veterans that have moved from the MIDDLE to TOP risk group
* Verify that the model identifies those Veterans in the TOP risk group that were first identified more than 30 days ago and have not yet had outreach efforts initiated to them (via their outreach status on the Roster Widget)
* Verify that the model identifies Veterans who received care at facilities that are NOT associated with the same VA Medical Center (VAMC) within the past year
* Verify that the model identifies the number of Veterans who received care at the same facility. Health professionals should receive Veteran information in messages sorted by location
* Verify that the application identifies the number of Veterans who received care at a facility and that facility has multiple recipients listed for DMs.

## User Functional Testing

Functional Testing is based upon the black box technique, defined as verifying the applications internal processes by interacting with the application via the Graphical User Interface (GUI) and analyzing the output or results.

This testing verifies functional performance and reliability requirements placed on major design items; these "design items" that are exercised through their interfaces.

Test Cases are constructed to test that all components interact correctly; for example, across procedure calls or process activations; this is done after testing individual modules (that is, Unit Testing).

The overall idea is to use a building block approach which is to continually test during each Sprint as stories are completed.

Table 4: Test Types

| Test Types | Party Responsible |
| --- | --- |
| Unit Testing | Developer |
| Compliance testing | Tester |
| Data and database integrity testing | Developer |
| Regression test | Tester/Developer |
| Section 508 compliance testing | Tester |
| System testing | Tester |
| Usability testing | Tester |
| User Functional Testing | Tester |
| Error Analysis Testing | Tester |

# Testing Techniques

A variety of testing techniques will be used, such as:

* Manual testing using Internet Explorer (IE), Safari, Chrome, and Firefox
* Automated testing uses a Ruby implementation of Cucumber integrated with the Selenium framework.
* Error testing through manipulation of environment components to simulation and generation of errors.

## Risk-based Testing

Risk-based testing will focus on the most precarious aspects of the system:

* Roles for the different type of dashboard users.
* Secure messaging
* Login into the application

## Break Testing

Break testing is performed with the intention of identifying weak points in the system. Break testing helps identify areas of concern and system vulnerabilities that would not otherwise be found by only testing actions within the ‘happy path’. Break testing will be completed on an iterative basis in order to identify bugs as development continues. Break testing will be conducted using manual processes. When feasible, automated break testing will be implemented using Jenkins and other testing tools listed in section [5.7. Productivity and Support Tools](#_Productivity_and_Support).

## Stress Testing

Stress testing is conducted to identify weak points within the application that could lead to system interruptions or break downs over time if left untreated. This testing methodology exacerbates regular user actions that fall within the ‘happy path’ to see how the application reacts. For example, this methodology would test 1500+ characters being entered into a text field. Stress testing will be conducted manually. When feasible, automated stress testing will be implemented using Jenkins and other testing tools listed in section [5.7. Productivity and Support Tools](#_Productivity_and_Support).

## Corner Case Testing

The objective of corner case testing is to test system user actions that are uncommon but possible. Corner case testing avoids testing actions that fall within the application’s ‘happy path’. The ‘happy path’ includes expected user application interactions that are required in order to use the system as designed. Corner case actions include unexpected scenarios that are possible but occur on a rare basis. It is expected that corner case testing will be conducted using manual testing processes and methodologies. When feasible, automated corner case testing will be implemented using Jenkins and other testing tools listed in section [5.7. Productivity and Support Tools](#_Productivity_and_Support).

**Happy Path** – The happy path includes predictable actions that a user may take when using an application. For the IRDS system specifically, user actions in the happy path can include:

* Logging into the system
* Restoring default widgets
* Switching between Dashboard views

**Corner Case** – Corner cases include less predictable actions that are possible when users interact with the system. For the IRDS system specifically, corner case actions can include:

* Resizing widgets to extremely large sizes (i.e. taking up the entire page)
* Selecting buttons multiple times (i.e. 10 times, 50 times, etc.)

**Happy Path**

**Corner**

**Case**

**Corner**

**Case**

Figure : Corner Case Testing

## Performance Testing

Performance testing will be conducted to understand how well the application reacts when users interact with the system. For example, performance testing could test the following items:

* Data Load Time – The length of time it takes for data to load on the webpage.
* Webpage Load Time – The length of time it takes for the page to load after logging in to the system (or when switching between views in the system).
* Widget Reaction Time – The reaction time when the user resizes and moves widgets. (i.e. Is there a lag when the user tries to move or resize a widget?).

Performance testing will be completed on an iterative basis in order to identify bugs and areas for improvement as development continues. Automated performance testing efforts will be implemented using the Apache JMeter testing tool. When automation is not feasible, manual testing methods will be used.

## Enterprise Testing

### Security Testing

If required a security test plan will be derived from the security specification section of the RSD. The objective of the tests is to validate the security requirements and to ensure readiness for the independent testing performed by VA’s Security Assessment Team per the Test and Certification Process. The security test plan will incorporate a vulnerability assessment. The security testing and vulnerability assessment will be conducted by the contractor test team per the discretion of the VA PM, the VA Information Security Officer (ISO), and other VA security team. Additional security testing and assessments including National Security Operations Center (NSOC) scans will be performed as noted in Appendix A: Office of Cyber Security (OCS).

See <https://vaww.portal2.va.gov/sites/infosecurity/ca/default.aspx> for additional Assessment and Authorization support services provided by the OCS.

### Privacy Testing

If required, a privacy test plan will be derived from the usability specification section of the PWS. The privacy test plan, if required, will be executed by the contractor test team and per the discretion of the VA PM, the VA Privacy Service, and a VA Privacy Officer.

A Privacy Impact Assessment (PIA) Validation letter will initiate this determination step in coordination with VA security and privacy representatives. A PIA will be completed in conjunction with the required Certification and Accreditation process. The PIA provides the following information:

* System details (name, owners, Federal Information Security Management Act [FISMA] data, etc.)
* System of record information including system of record notice (SORN) references
* Data collection and storage
* Data sharing and access
* Records management
* Security

VA Privacy Service information can be found at <http://vaww.privacy.va.gov/> .

### Section 508 Compliance Testing

Section 508 Testing will be handled in two phases, preliminary testing and formal compliance testing.

Preliminary testing will be performed by the contractor test team and development team via screen readers (i.e. Job Access with Speech [JAWS]) and keyboard-only testing techniques (no mouse). A Section 508 test plan will be created to direct these testing activities. The VA’s Section 508 Checklist for §1194.21 will serve as the basis for the test plan. An excerpt of the Section 508 Checklist for §1194.21 is shown in the table below. Upon successful completion of the preliminary testing the Section 508 Checklist for §1194.21, the Section 508 Conformance Validation Statement (CVS) will be submitted to the Section 508 Program Office (Section508@va.gov).

The goal of preliminary testing is to identify and fix section 508 compliance issues within the development phase to minimize the amount of subsequent rework and identify problem areas that would be raised by the VA’s Section 508 Certification Group. Section 508 compliance testing includes, but is not limited to:

* Screen Readers: All screens/fields are readable by a screen reader, for unsighted users
* Keyboard Accessibility: All screens are keyboard accessible (users can complete the screens without the use of a mouse). Verification that the tab order is logical and that the unsighted user can fully use the application in the same manner (with full functionality) as a sighted user

Table 5: Excerpt Section 508

| **Excerpt: Section 508 Checklist for §1194.21 Software Applications and Operating Systems** |
| --- |
| When software is designed to run on a system that has a keyboard, product functions shall be executable from a keyboard where the function itself or the result of performing a function can be discerned textually. |
| Applications shall not disrupt or disable activated features of other products that are identified as accessibility features, where those features are developed and documented according to industry standards. Applications also shall not disrupt or disable activated features of any operating system that are identified as accessibility features where the application programming interface for those accessibility features has been documented by the manufacturer of the operating system and is available to the product Developer. |
| A well-defined on-screen indication of the current focus shall be provided that moves among interactive interface elements as the input focus changes. The focus shall be programmatically exposed so that assistive technology can track focus and focus changes. |
| Sufficient information about a user interface element including the identity, operation and state of the element shall be available to assistive technology. When an image represents a program element, the information conveyed by the image must also be available in text. |
| When bitmap images are used to identify controls, status indicators, or other programmatic elements, the meaning assigned to those images shall be consistent throughout an application's performance. |
| Textual information shall be provided through operating system functions for displaying text. The minimum information that shall be made available is text content, text input caret location, and text attributes. |
| Applications shall not override user selected contrast and color selections and other individual display attributes. |
| When animation is displayed, the information shall be displayable in at least one non-animated presentation mode at the option of the user. |
| Color coding shall not be used as the only means of conveying information, indicating an action, prompting a response, or distinguishing a visual element. |
| When a product permits a user to adjust color and contrast settings, a variety of color selections capable of producing a range of contrast levels shall be provided. |
| Software shall not use flashing or blinking text, objects, or other elements having a flash or blink frequency greater than 2 Hertz (Hz) and lower than 55 Hz. |
| When electronic forms are used, the form shall allow people using assistive technology to access the information, field elements, and functionality required for completion and submission of the form, including all directions and cues. |
| \*§1194.31 Functional Performance Criteria  (a) At least one mode of operation and information retrieval that does not require user vision shall be provided, or support for assistive technology used by people who are blind or visually impaired shall be provided.  (b) At least one mode of operation and information retrieval that does not require visual acuity greater than 20/70 shall be provided in audio and enlarged print output working together or independently, or support for assistive technology used by people who are visually impaired shall be provided.  (c) At least one mode of operation and information retrieval that does not require user hearing shall be provided, or support for assistive technology used by people who are deaf or hard of hearing shall be provided.  (d) Where audio information is important for the use of a product, at least one mode of operation and information retrieval shall be provided in an enhanced auditory fashion, or support for assistive hearing devices shall be provided.  (e) At least one mode of operation and information retrieval that does not require user speech shall be provided, or support for assistive technology used by people with disabilities shall be provided.  (f) At least one mode of operation and information retrieval that does not require fine motor control or simultaneous actions and that is operable with limited reach and strength shall be provided. |

Formal compliance testing and approvals will be achieved through the VA Section 508 Program Office (<http://vaww.section508.va.gov/Section_508_Checklist_and_CVS.asp>). Per the VA Section 508 website “*In order to achieve conformance with the requirements of Section 508, a* VA PM must submit a documentation package for all E&IT products, services and documents. This package includes a completed Conformance Validation Statement. This is submitted with the products and services to the VA’s Section 508 Program Office, Standards Compliance Division in order to assure validation conformance. In addition, the PM must coordinate with the Department of Veterans’ Affairs Section 508 Program Office Testing and Training Center (T&TC) for product testing against the relevant standard. After testing, the T&TC staff will provide a copy of the CVS document signed by the VA Section 508 Coordinator to the PM acknowledging that the Electronic and Information Technology (IT) system meets the Section 508 requirements.”

## Productivity and Support Tools

The table below describes the tools that will be employed to support this MTP:

Table 6: Tool Category or Types

| Tool Category or Type | Tool Brand Name | Vendor or In-house | Description |
| --- | --- | --- | --- |
| Test Management | Jira | Vendor | A web application used for defect and project management. |
| Defect Tracking | Jira | Vendor | A web application used for defect and project management. |
| Project Management | Jira | Vendor | A web application used for defect and project management. |
| Functional Test Automation | Cucumber  Selenium  Ruby  Open Web Application Security Project (OWASP) | Open Source | A suite of test tools, including support for web browser automation. |
| Performance Testing | Apache JMeter | Open Source | A 100% pure Java desktop application, designed to load test functional behavior and measure performance. |
| Unit Testing | Jasmine | Open Source | A framework for testing JavaScript code. |
| Miscellaneous Software and Tools | VA Innovation Sandbox Virtual Machine (VM)  VirtualBox | Open Source  VA  Oracle | A cloud-based, virtual space for developing and testing applications |
| 508 Section Testing | Web Accessibility Evaluation Tool (WAVE) | Open Source | WAVE is a tool that aids web developers in making their web content more accessible. |
| 508 Section Testing | PowerMapper | Open Source | PowerMapper is a software that makes easy-to-use tools for mapping, testing and analyzing web applications. |
| 508 Section Testing | NonVisual Desktop Access (NVDA) | Open Source | NVDA is a “screen reader” which enables blind and vision impaired people to use Application. It reads the text on the screen in a computerized voice |
| 508 Section Testing | Manual Keyboard Test (No mouse) | N/A | Without the aid of a mouse being able to navigate the webpage with only the keyboard. |

# Criteria

## Pass/Fail Criteria

Assessments of pass or fail criteria are made for each Test Case and Test Script. The following guidelines are provided to assist all parties involved in testing.

A Priority Level is also assigned to rank the order with which the defects are addressed. There are five Priority Levels: Blocker, Critical, Major, Minor, and Trivial.

Table 7: Pass/Fail Guidelines

| Condition | Guideline |
| --- | --- |
| **Pass (P)** | Results of the action agree with what is expected |
| **Pass with Constraints** | Results do not completely agree with expected results, but the results are just a slight inconvenience to the user (for example not the preferred number of results or mislabeled headers). |
| **Fail (F)** | The function has failed, and there is no workaround; or an action results in a system crash or results in damage to data (for example, incorrect results or the loss of data). |

Test results which are reviewed contain subsystem Test Cases and the corresponding results which were collected during execution. Each system test results are reviewed during the Sprint with the Development Team and collectively at the end of each Sprint in the Sprint Planning meeting.

## Acceptance Criteria

The acceptance criteria listed below are used to determine whether a feature satisfies the acceptable level of quality.

* All tests are running green on the CI server
* All tests meet the acceptance criteria of the story/feature
* All critical bugs have been resolved and validated

# Test Deliverables

The table below lists the test deliverables for the IRDS Project:

Table 8: Test Deliverables

| Test Deliverables | Responsible Party |
| --- | --- |
| Master Test Plan | Test Lead, Test Engineer |
| Test Frameworks | Test Lead, Test Engineer |
| Testing Manual | Test Lead, Test Engineer |
| Test Cases/Test Scripts | Test Lead, Test Engineer |
| Test Data | Test Lead, Test Engineer, Development Team |

# Test Environment

The VA Innovation Sandbox will be used to develop and test the new software. The VA Innovation Sandbox is a cloud-based, virtual space for developing and testing applications. 

**Figure 3: Test Environment**

## Base Software Elements in the Test Environments

The table below describes the base software elements that are required in the test environment for this MTP:

Table 9: Software Elements

| Software Element Name | Type and Other Notes |
| --- | --- |
| VA Innovation Sandbox | A cloud-based, virtual space |
| MS SQL Server | Reach Database |
| R and BIRT | Data Analytics Tools |
| Angular JS, NodeJS, Express.js, HTML, JavaScript, and CSS, IIS8 | Dashboard Deployment |
| Selenium, Ruby, Cucumber, WAVE, DbUnit, Apache JMeter, and Jenkins | Testing and Build tool |

# 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 defects identified; listed by cause and severity

# Acronyms

Table 10: Acronyms

|  |  |
| --- | --- |
| Term | Definition |
| A&A | Assessment and Authorization |
| ATO | Authority to Operate |
| BIRT | Business Intelligence and Reporting Tools |
| CI | Continuous Integration |
| CM | Configuration Management |
| CPO | Certification Program Office |
| CPRS | Computerized Patient Record System |
| CVS | Conformance Validation Statement |
| DRT | Document Review Team |
| DM | Direct Message |
| ERM | Enterprise Risk Management |
| EUT | End User Testing |
| FISMA | Federal Information Security Management Act |
| GUI | Graphical User Interface |
| IE | Internet Explorer |
| SME | Subject Matter Expert |
| IIS | Internet Information Services |
| IRDS | Integrated Reach Database System |
| ISO | Information Security Officer |
| IT | Information Technology |
| JAWS | Job Access With Speech |
| MTP | Master Test Plan |
| NIST | National Institute of Standards and Technology |
| NSOC | National Security Operations Center |
| NTP | National Teleradiology Program |
| OCS | Office of Cyber Security |
| OMB | Office of Management and Budget |
| PIA | Privacy Impact Assessment |
| PM | Project Manager |
| PR | Perceptive Reach |
| PWS | Performance Work Statement |
| QA | Quality Assurance |
| RSD | Requirements Specification Document |
| RTM | Requirements Traceability Matrix |
| SCA | Security Control Assessment |
| SDD | System Design Document |
| SDP | Software Development Plan |
| SMART | Security Management and Report Tool |
| SORN | Systems of Record Notice |
| SP | Special Publication |
| SPC | Suicide Prevention Coordinator |
| SQA | Software Quality Assurance |
| T&TC | Department of Veterans’ Affairs Section 508 Program Office Testing and Training Center |
| TATO | Temporary Authority to Operate |
| TDD | Test Driven Development |
| TRM | Technical Reference Model |
| UI | User Interface |
| VA | Department of Veterans Affairs |
| VACI | Department of Veterans Affairs Center for Innovation |
| VHA | Veterans Health Administration |
| VLER | Virtual Lifetime Electronic Record Direct |
| VM | Virtual Machine |

## References

### Documentation, Resources, and Repositories

The following documentation, resources, and repositories support the development and implementation of this MTP:

* [VA Handbook 6500](http://vaww1.va.gov/vapubs/viewPublication.asp?Pub_ID638&FType=2)

<http://vaww1.va.gov/vapubs/viewPublication.asp?Pub_ID638&FType=2>

* [508 Compliance](http://vaww.section508.va.gov/index.asp)

<http://vaww.section508.va.gov/index.asp>

* [System Design Document (SDD)](https://github.com/VHAINNOVATIONS/PerceptiveReach)

<https://github.com/VHAINNOVATIONS/PerceptiveReach>

* [Software Development Plan (SDP)](https://github.com/VHAINNOVATIONS/PerceptiveReach)

<https://github.com/VHAINNOVATIONS/PerceptiveReach>

* [Technical Reference Model](https://www.voa.va.gov/) (TRM)

<https://www.voa.va.gov/>

* [Perceptive Reach Innovation Pipeline Home Page](http://vacloud.us/groups/558/)

<http://vacloud.us/groups/558/>

* [Perceptive Reach Requirements Specification Document (RSD)](https://internal.vacloud.us/wiki/projects/558/)

<https://internal.vacloud.us/wiki/projects/558/>

* [Perceptive Reach Requirements Traceability Matrix (RTM)](https://internal.vacloud.us/wiki/projects/558/)

<https://internal.vacloud.us/wiki/projects/558/>

# Appendix A

## A.1 Office of Cyber Security

The Office of Cyber Security (OCS) supports security Assessment and Authorization (A&A) functions as required by The Office of Management and Budget (OMB) based on guidance formulated by the National Institute of Standards and Technology (NIST).

Security Assessment and Authorization is part of the FISMA Implementation Project. This Implementation Project, as provided for by NIST, “promotes the development of key security standards and guidelines to support the implementation of and compliance with the Federal Information Security Management Act.”

The Certification Program Office (CPO) in OCS implements programs and projects designed to support FISMA compliance. The CPO gains its authority, in part, from VA Handbook 6500.3. The functional areas of the CPO are as follows:

* Security Control Assessment (SCA)
* Authorization Support (formerly Accreditation Support)
* Security Management and Report Tool (SMART)

April 30, 2013

* GovInfoSecurity: NIST Unveils Security, Privacy Controls (NIST SP 800-53 rev 4 updated)
* http://www.govinfosecurity.com/interviews/nist-unveils-security-privacy-controls-i-1907
* NIST Special Publication (SP) 800-53: Security and Privacy Controls for Federal Information Systems and Organizations updated
  + Impacts A&A as internal VA OIS SSP & SCA templates will need to be updated with more controls in the coming months - this is just a heads-up so that those involved with any responsibility of A&A of their system, they will have to be in compliance of the new NIST guidance come May 1, 2014. So it is suggested that key stakeholders start getting familiar with this latest guidance released by NIST
* Related GCN Article:
  + “The changes are substantial,” said Ron Ross, the FISMA implementation lead at NIST. “The fundamental underpinnings haven’t changed,” but the catalog of security controls has grown from more than 600 to more than 850 controls, and there is a new emphasis on the underlying trustworthiness of systems and on privacy controls.”

Early 2013

Temporary Authority to Operate (TATO) and ATO Extensions approved

* For all systems that had a valid TATO or ATO, whose TATO or ATO expired late 2012 or before August 31, 2013, will have their TATO or ATO automatically extended through August 31, 2013. By the 31st of August, we hope to have the GRC tool operational, called RiskVision, and have its functions operational that will address A&A activity. Guidance regarding the tool, implementation, and training will be forth coming in the near future.

New Systems requesting TATO or ATO for 2013

Until the GRC tool is officially operational, the following A&A activity will continue.

* Security Requirements – comply with required NIST and agency controls
* Identify ISO to align with this system (An ISO will be provided once you send in your email as documented below). You will not know the NISO until you have an ISO assigned. (FFS ISO Support Request.pdf)
* Pick an official name for the system, then get with the SMART Help Desk (point of contact Lizbeth L. Gray), and submit the attached SMART System Update Form to request for/then establish a record in SMART of this system. (SMART System Update Form.doc)
* Identify the required security documentation necessary to ensure controls/NIST and agency security requirements are addressed (For the latest template – your ISO or Martin DeLeo would be your best POC)
  + - SCA Signatory Authority:
    - System Security Plan
    - Risk Assessment
    - PIA
    - Contingency Plan
    - Disaster Recovery Plan
    - Incident Response Plan
      * Configuration Management Plan
    - SCA Security Configuration Checklists
    - Interconnected Systems Agreements
* Have ISO coordinate the development/review/approval of the noted documents above (reminder – your ISO is your custodian/support point of contact, but not the author of these documents)
* Once approved - upload each document in to SMART database (via SMART – Artifacts)
* Contact the Certification Program Office for coordinating the A&A activities (you would complete and submit your Request for ATO memo (Request for an ATO Memo.docx) to Steven Kish, which will start the ATO review process – be sure when identifying your system that you use the full formal system name as it resides in SMART)
  + - Assessment
    - Accreditation
* Your required security documentation will be required to undergo a formal review by the document review team (DRT)
* You will be required to identify which scan or scans applies to your system, complete the scan request form(s) then submit it/them to the NSOC POC with a cc to Steven Kish
* The Enterprise Risk Management (ERM) team may be notified of your system and you will work with them to get an on-site SCA scheduled and conducted
* Deficiencies from the DRT review, NSOC scan(s) and SCA site visit will be shared and Plan of Action and Milestones (POA&Ms) will need to be established and remediation efforts initiated
* Once your system’s level of deficiencies reach an acceptable level, your system will be packaged and submitted to OSC management’s approval with a recommendation to grant either a Temporary ATO or a full ATO

As you can see, it starts by getting an ISO identified and assigned and then placing this system into the SMART database. Once established in SMART, the Team then generates/develops the required security documents so they can then be uploaded into SMART – Artifacts section. All systems that need FISMA reporting must be established in SMART and have their security documents uploaded within. To start the process in getting an ISO assigned to your system, you would submit your request and e-mail it to: VAFSSISORequests@va.gov

The information being requested is key to the establishment of any system in SMART. In order to request an ISO, please complete the ISO request form to describe the type of information security officer support required. State who you are and what project you are supporting and that you are formally requesting ISO support for your Initiative. Ensure that you include as much of the following information as possible:

* System Owner
* Business Owner
* Performance Work Statement
* If in test phase give sites; what are sites that will ultimately use initiative
* Developer name
* And anything else that is pertinent.

With regards to NSOC scans:

Please see which applies and any that does will need to be completed and submitted as part of the security control assessment/testing process. Your completed form(s) will need to be sent to James Barrett and/or Jason Kendall with a CC to Steven Kish.

Below are the 3 NSOC documents with a short description of each. When filling out one or more forms, if you have questions – please contact James Barrett and/or Jason Kendall.

* Scan request form (VA Scanning Request.pdf) – This request form will be used when vulnerability scans are requested for workstations, servers, LAN’s, etc. James Barrett is the POC for this task.
* WASA questionnaire (WASA Questionnaire v.10.docx) – This request form will be used when a web application is present. Jason Kendall is the POC for this web application penetration testing.
* Source code review questionnaire (Code Review Questionnaire v 1.4.docx) – This questionnaire will be used when a custom application has been created (not proprietary COTS). Jason Kendall is the POC for this source code testing.