Perceptive Reach

Integrated Reach Database System

(IRDS)

System Administration Guide



Department of Veterans Affairs

**September 2016**

**Version 2.0**

Revision History

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| 9/13/2016 | 2.0 | Management Review | Matthew Robinson |
| 09/11/2016 | 1.8 | Added additional database side system administration content. | Bill Balshem |
| 09/09/2016 | 1.7 | Added Section 2.1.1 | Vasudha Upadhyaya |
| 5/23/2016 | 1.6 | Management Review | Matthew Robinson |
| 3/28/2016 | 1.5 | Updated the title page, page headers, and page footers to reflect the correct month  Updated section “1.4.3. Dependent Systems” to include two systems that were previously not listed. | Kaitlin Reskovac |
| 2/16/2016 | 1.4 | Updated page headers and footers to reflect the correct month. | Kaitlin Reskovac |
| 1/27/2016 | 1.3 | Added table caption in section “5. Acronyms & Abbreviations” | Kaitlin Reskovac |
| 1/20/2016 | 1.2 | Updated headers and footers to reflect the correct month and year. | Kaitlin Reskovac |
| 12/10/2015 | 1.1 | Option Year Review & Update | Kaitlin Reskovac |
| 09/14/2015 | 1.0 | Final Review | Paul Bradley/Monica Mohler |
| 09/08/2015 | .6 | Contract Compliance and Quality Assurance Review | Matthew Robinson/Kaitlin Reskovac/Radina Ivanova |
| 09/08/2015 | 0.5 | Added content to database automated processes sections | Bill Balshem |
| 09/04/2015 | 0.4 | Document updates | Andal FeQuiere |
| 07/14/2015 | 0.3 | Document updates | Andal FeQuiere |
| 06/12/2015 | 0.2 | Document updates | Andal FeQuiere/Bill Balshem |
| 05/10/2015 | 0.1 | Initial Version | Matthew Robinson/ Radina Ivanova |

Table of Contents

[1. System Business and Operational Description 1](#_Toc461564581)

[1.1. Operational Priority and Service Level 1](#_Toc461564582)

[1.2. Logical System Description 1](#_Toc461564583)

[1.3. Physical System Description 3](#_Toc461564584)

[1.4. Software Description 4](#_Toc461564585)

[1.4.1. Background Processes 5](#_Toc461564586)

[1.4.2. Job Schedules 5](#_Toc461564587)

[1.4.3. Dependent Systems 5](#_Toc461564588)

[2. Routine Operations 6](#_Toc461564589)

[2.1. Administrative Procedures 6](#_Toc461564590)

[2.1.1. System Start-up 6](#_Toc461564591)

[2.1.2. System Shut-down 7](#_Toc461564592)

[2.1.3. Back-up & Restore 7](#_Toc461564593)

[2.1.3.1. Storage and Rotation 7](#_Toc461564594)

[2.2. Database Automated Processes 7](#_Toc461564595)

[2.2.1. Patient Risk Score Run 7](#_Toc461564596)

[2.2.1.1. Patient Days Use Import SSIS package 8](#_Toc461564597)

[2.2.1.2. CDW Import SSIS package 11](#_Toc461564598)

[2.2.1.3. Patient Primary VAMC SSIS package 11](#_Toc461564599)

[2.2.1.4. Patient Risk Scoring SQL stored procedure 12](#_Toc461564600)

[2.2.1.5. Patient Dashboard Stratification SQL script 12](#_Toc461564601)

[2.2.1.6. Clinical Support Import SSIS package 12](#_Toc461564602)

[2.2.1.7. Suicide Data Repository (SDR) Import SSIS package 12](#_Toc461564603)

[2.2.1.8. VLER Direct Message Node JS process 13](#_Toc461564604)

[2.2.2. Facility Surveillance Processes 13](#_Toc461564605)

[2.2.2.1. Facility Surveillance Backend Process 13](#_Toc461564606)

[2.2.2.2. VAMC Year End Enrollment SSIS package 13](#_Toc461564607)

[2.3. Security / Identity Management 14](#_Toc461564608)

[2.3.1. Identity Management 14](#_Toc461564609)

[2.4. Dashboard Management 14](#_Toc461564610)

[2.5. User Notifications 15](#_Toc461564611)

[2.6. System Monitoring, Reporting & Tools 15](#_Toc461564612)

[2.6.1. Availability Monitoring 15](#_Toc461564613)

[2.6.2. Performance/Capacity Monitoring 15](#_Toc461564614)

[2.6.3. Critical Metrics 16](#_Toc461564615)

[2.7. Routine Updates, Extracts and Purges 16](#_Toc461564616)

[2.8. Capacity Planning 17](#_Toc461564617)

[3. Exception Handling 18](#_Toc461564618)

[3.1. Routine Errors 18](#_Toc461564619)

[3.1.1. Security Errors 18](#_Toc461564620)

[3.1.2. Time-outs 19](#_Toc461564621)

[3.1.3. Concurrency 19](#_Toc461564622)

[3.2. Significant Errors 20](#_Toc461564623)

[3.2.1. Application Error Logs 20](#_Toc461564624)

[3.2.2. Application Error Codes and Descriptions 20](#_Toc461564625)

[3.2.3. Infrastructure Errors 21](#_Toc461564626)

[3.2.3.1. Database 21](#_Toc461564627)

[3.2.3.2. Web Server/Application Server 21](#_Toc461564628)

[3.2.3.3. Network 22](#_Toc461564629)

[3.2.3.4. Authentication & Authorization 22](#_Toc461564630)

[3.3. Dependent System(s) 22](#_Toc461564631)

[3.4. Troubleshooting 22](#_Toc461564632)

[3.5. System Recovery 24](#_Toc461564633)

[3.5.1. Restart after Non-Scheduled System Interruption 25](#_Toc461564634)

[3.5.2. Restart after Database Restore 25](#_Toc461564635)

[3.5.3. Back Out Procedures 25](#_Toc461564636)

[4. Operations & Maintenance System Support 26](#_Toc461564637)

[4.1. Support Structure 26](#_Toc461564638)

[4.1.1. Support Hierarchy 26](#_Toc461564639)

[4.2. Support Procedures 26](#_Toc461564640)

[5. Acronyms & Abbreviations 27](#_Toc461564641)

# System Business and Operational Description

This System Administration Guide (SAG) 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 to mitigate the development of risk.

## Operational Priority and Service Level

IRDS’s overall importance to VA lies within its ability to augment current suicide prevention programs with supporting data for proactive, “upstream” suicide prevention approaches. To do this, the application applies sophisticated predictive analytics frameworks to Veteran health data. The application of these frameworks produces stratified lists of Veterans who are potentially at an increased risk for suicidal ideation or attempt. Clinical providers, such as VA Suicide Prevention Coordinators (SPCs) can then use this information to reach out to Veterans. The customers served are the Veterans who consume VHA-provided services, VHA clinicians, and supervisors. In addition, parties interested in research and reporting based on the data held in the IRDS will be granted special access to perform these tasks.

IRDS is considered *important*, but not *critical* for patient care, in most cases. If the system is down or otherwise unavailable, patients’ safety will not be immediately compromised and standard VA programming for suicide prevention will be available as a contingency. However, the system’s unique ability to identify and report Veterans at an increased risk for suicidal ideation or other behavioral health issues in real time implies that the service should be prioritized above other non-critical systems.

## Logical System Description

The Perceptive Reach program, including components of the IRDS, are as follows:

* Reach Database – a Structured Query Language (SQL) database (DB) storing data used for analytic input
* Data Analytics Platform – an integrated collection of analytic tools
* Risk Model – the predictive model(s) used to identify high-risk Veterans
* Dashboard – a multi-view information portal displaying results from the analytic platform and risk model
* Direct Messaging – a secure messaging solution to notify outreach and intervention resources/clinicians about high-risk Veterans

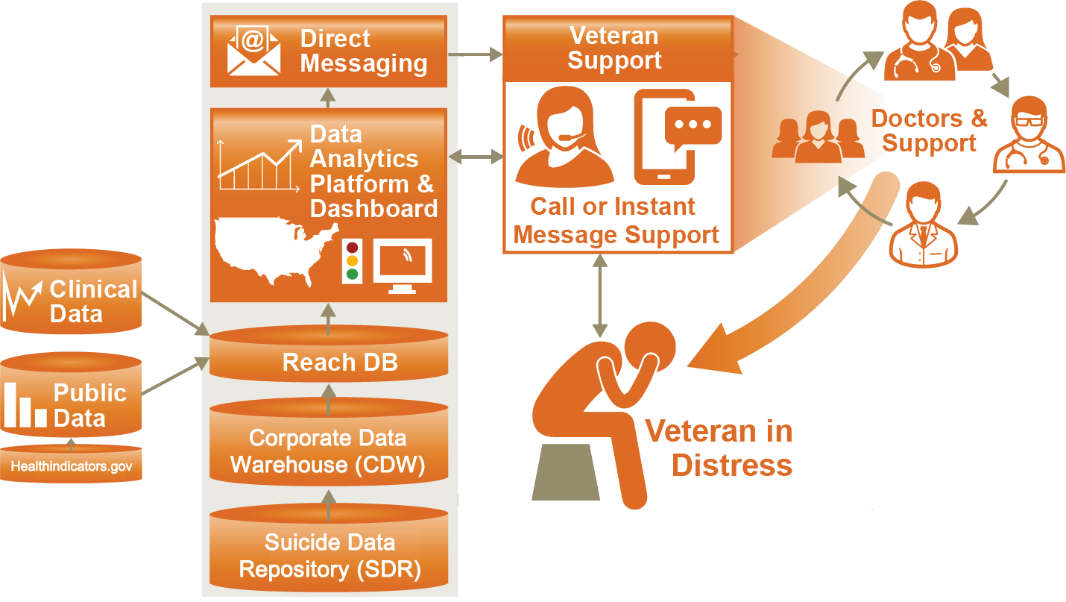


Figure 1: IRDS System Overview

The primary ingress interfaces will use SQL Server Integration Services (SSIS) to retrieve data from the designated data sources such as the Corporate Data Warehouse (CDW) and Suicide Data Repository (SDR). The SSIS package will be configured to pull data from the designated data source at a specified interval (daily/weekly/monthly/annually). The primary egress interface will be the Direct Messaging interface used to notify VA SPCs and other frontline care coordinators and clinicians. The IRDS will leverage the Virtual Lifetime Electronic Record (VLER) Direct messaging service utilizing a RESTful interface to transmit message content. Another interface utilized by VA Staff and clinical providers such as SPCs will be the IRDS Surveillance Dashboard which will provide browser based visualizations of critical data to identify at-risk Veterans.



Figure 2: IRDS Interface Overview

## Physical System Description

| **Type** | **OS** | **Processor** | **Memory** | **Storage** | **Users** | **Applications** |
| --- | --- | --- | --- | --- | --- | --- |
| Cloud (Database Server) | MS Windows Server 2012 64-bit | Intel Xeon CPU X7560, 2.27GHz  4 processors | 64 GB | 480 GB | 16 | MS SQL Server 2012 Enterprise Edition |
| Cloud (Application Server) | MS Windows Server 2012 64-bit | Intel Xeon CPU X7560, 2.27GHz  4 processors | 64 GB | 180 GB | 16 | IIS 7.0 |

The IRDS will be hosted on 2 IT Pilot servers under the control of Austin Information Technology Center (AITC).

Table 1: System Hardware

## Software Description

The system centers on the Reach Database as depicted in **Figure 3**. Veteran data flows in from VA sources, initially CDW, and is processed through a Risk Assessment which categorizes each Veteran at a risk level. Based on gathered requirements, the IRDS Dashboard and Direct Message notification process will interact with Veteran data to provide support through visualizations and notifications to VA SPCs.

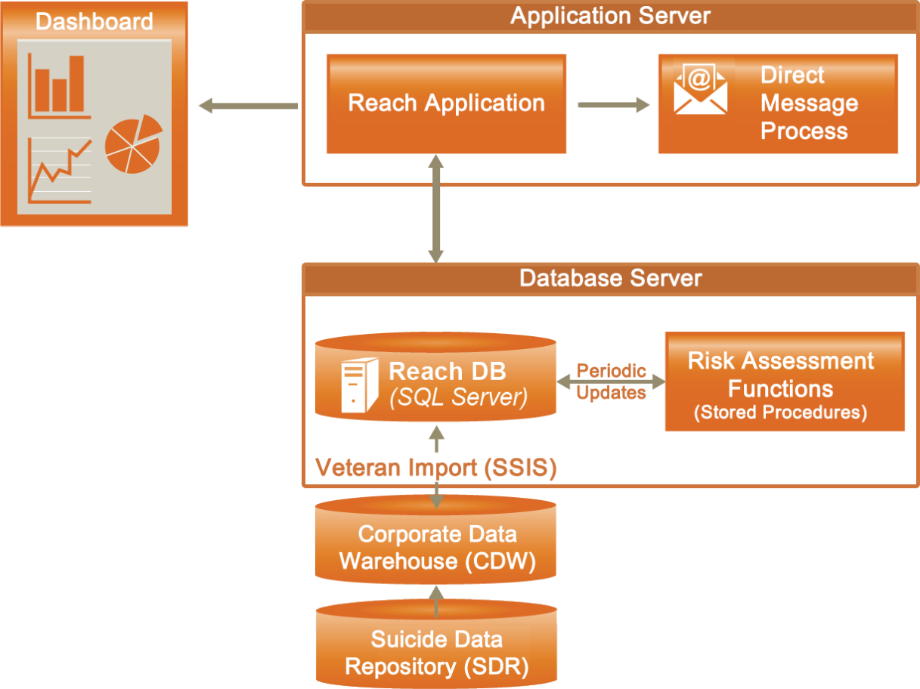


Figure 3: Application Design

A full list of the software used in the system is described below:

| Name | Technologies | License |
| --- | --- | --- |
| Dashboard | HTML5, CSS3, JavaScript, JQuery, Angular JS, Node.js | Open source |
| Direct Message Assembler | Node.js, JavaScript, VLER Direct API | VA |
| Data Analytics Platform | R Studio, BIRT | Open source |
| Reach DB | SQL Server 2012 | Commercial (provided) |
| Web Server | IIS, Windows Server 2012 | Commercial (provided) |

Table 2: Software Used in IRDS

All application software is open source or provided by the VA. The operating system is Windows due to the dependency on using SQL Server.

### Background Processes

The application background processes are as follows:

* InetMgr.exe: The IIS application which hosts the web application
* sqlservr.exe: The server daemon for the SQL Server database
* node.exe: The JavaScript runtime technology to host our server side application

**Note:** SQL Server is configured to run as a Windows service so that it starts automatically with Windows.

### Job Schedules

Schedule jobs are defined in detail in [Section 2.2](#_Database_Automated_Processes).

### Dependent Systems

| **Name** | **Description** | **Interface Name** | **Interface System** |
| --- | --- | --- | --- |
| VLER Direct messaging | VA proprietary email like messaging system for notifications that contain PII/PHI. The IRDS system notifies Suicide Prevention Coordinators (SPCs) when patients tied to their facility have been identified as at risk for attempting suicide. | HTTPS RESTful API via Node.js | Reach DB, VLER Direct |
| Corporate Data Warehouse (CDW) | Warehouses VistA data in SQL format. | SSIS | Reach DB |
| VA Suicide Data Repository (SDR) | Warehouses multiple VHA and external data sources such as National Death Index (NDI) mortality data. | SSIS | Reach DB |
| MedSAS | Warehouses VistA data in SAS Format. | SAS, text files | Reach DB |

Table 3: Dependent Systems

# Routine Operations

There are two routine operations that must be performed on the system: user management and backing up the database. User management is performed within the IRDS DB using SQL Server Management Studio (SSMS) and consists of adding, editing, and removing users, and is covered in [Section 2.2 Security/Identity Management](#_Security_/_Identity). Backing up the database is performed at the operating system level and is covered in [Section 2.1.3 Back-up & Restore](#_System_Recovery).

| **Operation** | **Role** | **Section** |
| --- | --- | --- |
| User management | Program technical administrator | 2.2 |
| Database backup | System administrator | 2.1.3 |

Table 4: Routine Operations

## Administrative Procedures

### System Start-up

The system does not require any regular manual start-up procedures. The database and application servers are both implemented as Windows services that automatically start with Windows, and it is unlikely that either service will fail under normal conditions. The services are listed below:

| **Service** | **Name** | **Display Name** |
| --- | --- | --- |
| SQL Server (MSSQLSERVER) | MSSQLSERVER | MSSQLSERVER |
| Windows Process Activation Service (IIS) | WAS | WAS |
| World Wide Web Publishing Service (IIS) | W3SVC | W3SVC |

Table 5: System Services

In the event that either does not start, or shuts down prematurely, they can be manually started via the Windows services snap-in. However, before starting either service, consult the Windows event viewer and individual service logs for information about the error. See [Section 3 Exception Handling](#_Exception_Handling) for more details about error handling and logs.

### System Shut-down

The system can be shut down by shutting down the two system processes from the Services snap-in. The services are listed in [Section 2.1.1 System Start-up](#_System_Start-up).

### Back-up & Restore

Database backup and restoration is performed and managed by the AITC staff assigned to the IRDS system. The IRDS development team will communicate to the AITC staff which database on the IRDS database server need to be backed up. For those databases, AITC will perform hot backups of the IRDS production database “Reach” directly to tape on a nightly basis. For information on requesting a production database restore from tape backup see the IRDS deployment guide.

The IRDS development team has the option to create a backup to disk on the data (E:) drive when necessary, such as when updates are being applied to production database. In the event that an issue occurs. The database can quickly be restored from backup saved to disk.

#### Storage and Rotation

The VA system administrator should copy/move database backups to remote storage for safekeeping according to IT guidelines. The IRDS server provides RAIDed storage and a rolling backup system for safeguarding backups locally, but relocating backups to Storage Area Network (SAN) or other storage per VA IT guidelines provides additional safety and redundancy. External/redundant storage by IT is beyond the scope of this document.

## Database Automated Processes

### Patient Risk Score Run

NOTE: This section describes the database automated processes from an administrative standpoint. For more detail on these processes see the IRDS Developer’s Guide.

Patient risk scores are run every few weeks/months at the discretion of the IRDS data owner. The full process consists of multiple automated, but not scheduled processes, run one after another in succession, that run over the course of about 24-48 hours. The majority of the patient risk scoring processing happens on the Corporate Data Warehouse (CDW) server where patient data is analyzed to define which patients should be in the current IRDS population and calculates the risk model data points for those patients.

The IRDS user based should refrain from logging into the IRDS until these processes have completed running.

The risk score processes should be run over the weekend where traffic on the CDW server is lower and would generally be a period of non-working hours for the IRDS user staff.

The patient risk score run consists of the following processes

1. Patient Days Use Import SSIS package
2. CDW Import SSIS package
3. Patient Primary VAMC SSIS package
4. Patient Risk Scoring SQL stored procedure
5. Patient Dashboard Stratification SQL script
6. Clinical Support Import SSIS package
7. Suicide Data Repository (SDR) Import SSIS package
8. VLER Direct Message Node JS process

#### Patient Days Use Import SSIS package

One category of risk factors calculated for patients in the IRDS system is Days Use. This includes inpatient visits, inpatient mental health visits, outpatient visits, outpatient mental health visits and overall visits. These values are rolled up into 30 day aggregates for a 2 year period (ex: Inpatient visits 8 months prior). Because patient scoring in IRDS is always done on a current snapshot of patient data, these values are always changing. Initially the processing for these data points were done as part of the CDW import, but the processing would take 3-4 days to run for the entire IRDS population. Therefore a design change was made where patient days use data is tracked in tables in the IRDS database, that track patient usage every day for the past 730 days. When the CDW import is run it then calculates day usage variables by rolling up values from these tables, which runs in 1-2 hours.

A separate SSIS package was created to import days use data into the IRDS database for the patient population, one day at a time. The 1st version of the solution processed yesterday’s days use data and could be scheduled on a nightly basis. The issue with this is that if there is a problem with one or more runs and days(s) are missed, the integrity of data can be compromised (see the IRDS developers guide for more details).

Therefore the Days Use import is designed where periods of days use calculated in one run. The period is specified by the DBA, by placing entries into table located in a CDW database. The package calculates days use, 1 day at a time, for the entire period requested by the DBA. It takes about 15-20 minutes to process one day of patient usage. The management of this solution is at the discretion of the DBA. For example, the DBA can:

* At the end of each week run one weeks’ worth of day use
* Before patient risk scores are calculated run for the period since the last risk scores were run

To run the Days Use Import SSIS package:

1. Verify The last day processed

In SQL Server Management Studio (SSMS) run the following query against the production database:

Select value From fn\_listextendedproperty('Description', 'SCHEMA', 'dbo', 'TABLE', 'DaysUse\_IP', 'COLUMN', 'Day1')

This query will return the last day of patient usage that was calculated for IRDS population

ex: query returns 'Run on 2016-09-01'

Before the next SSIS package is run, days use should be calculated up to the day risk scores will be calculated. If the risk score process is going to be run on 9/15/2016, days use will need to be processed for 9/2 - 9/14

1. Request Days to Process

In SSMS connect to the database and select the project database, open a query panel and run the following script:

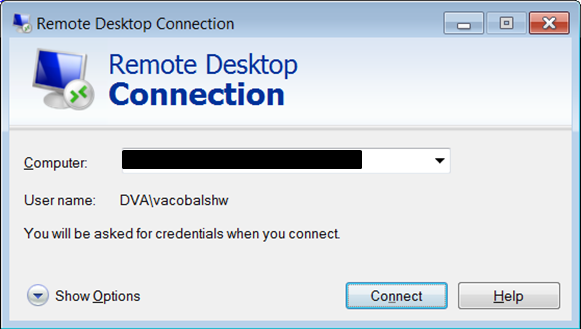
insert into staging.DaysToRun (RunDate) values ('09/02/2016')

Insert 1 record for each day requested to be processed. Using the sample above a record would be inserted for every day between 9/2/2016 through 9/14/2016

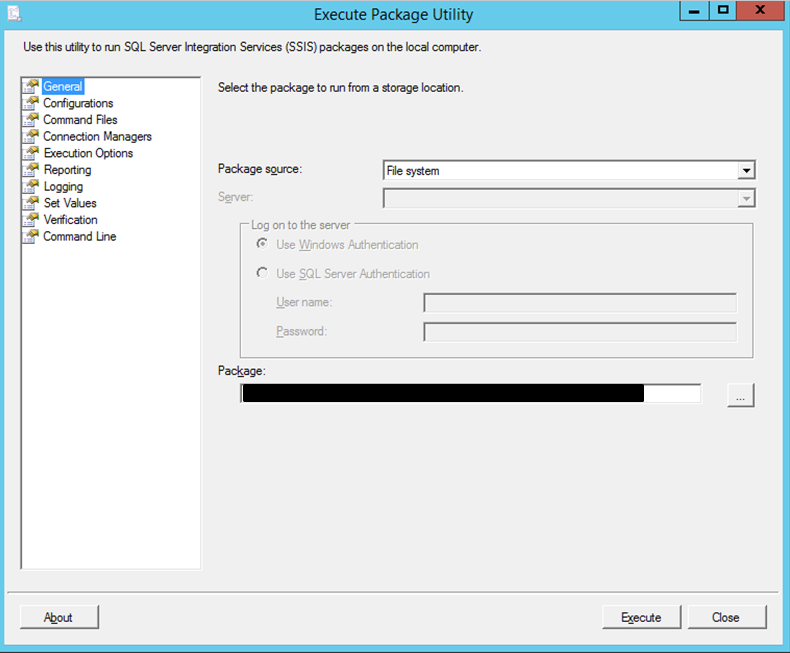
1. Run the package

NOTE: The development and deployment environments for the IRDS SSIS solutions reside on the CDW ETL server. This document assumes the DBA has gained the appropriate access to develop, deploy and execute all SSIS IRDS solutions. The detail on these environments is discussed in detail in the IRDS Developers guide.

1. Remote desktop to the CDW ETL server.



1. Go to the program executable folder and execute the SSIS package.



Reporting

Each SSIS package will create a report in the report folder.

* a completion report is created if the package runs successfully
* an error report is created if the an error occurs during the run

The naming convention for these report is:

[import\_name]\_[completion/error]Report\_[date/time run].txt

ex: if the package above ran at 1pm

* if it were successful would create the file DaysUse\_CompletionReport\_9\_14.2016\_0100.txt

this file just contains a start and end time

* if it failed would create the file DaysUse\_CompletionReport\_9\_14.2016\_0100.txt

this file contains a message of the error that caused the package to fail

Issue Research

If there is a need to review the SSIS code to resolve an issue that occurs, each SSIS project is located under the development folders. The project folder will have the same naming convention as the deployment folder. The Days Use SSIS solution would be located in the Visual Studio project folder.

#### CDW Import SSIS package

This package adds analyzed patient data from the CDW database to:

* Add new VHA users since the last run to the IRDS system
* Update the risk factors of the current IRDS patient population

To execute this package go the file and execute the package as per the instruction in section 2.2.1.1.

Override

CDW has between 7-8 million current users in the production database. It was determined that processing 1 million patients at a time was efficient. The import has 8 loops (as well as some front end/back end process before/after these loops). For more detail see the IRDS Developers guide. Each loop takes about 2-3 hours to run. If an issue occurs, such as a connection loss between the CDW/IRDS servers, the import does not need to be started from the begging. There is override functionality built into the system.

To override the import:

1. Open SSMS an connect to the CDW database
2. Select the IRDS project database, open a query window and run:

select count(\*) from ETL.PatientQueue

The count will be exact to the million, unless the final loop completed where the count would be 7M +

ex: query result returned 6M.

This means the package crashed during the 7th loop.

The package can be restarted at the beginning of the 7th loop so minimal processing time will be lost.

1. Go to the package configuration file and open in a text editor

Look for the value:<Configuration ValueType="Int64" Path="\Package.Variables[User::Loop\_CTR].Properties[Value]" ConfiguredType="Property"><ConfiguredValue>1</ConfiguredValue>

1. In the ConfiguredValue tag change, change the value from 1 to 7
2. Re-execute the IRDS\_CDW\_Import.dtsx package

**NOTE: Make sure to change the value in the ConfiguredValue tag back to 1 before the next risk score process is run**

#### Patient Primary VAMC SSIS package

This package reviews tables on the CDW server to find each patients current primary facility. The results are updated in the IRDS production database. Although the IRDS filters out patients who are flagged as deceased in CDW, patients might become deceased after being imported into IRDS. These patients are identified and flagged accordingly.

To execute this package go the file and execute the package as per the instruction in section 2.2.1.1.

#### Patient Risk Scoring SQL stored procedure

This stored procedure:

1. Scores patients based on their risk factors per the current risk model, where the higher the score, the more likely the patient is to attempt suicide
2. Identifies the risk levels for with the categories being the top .1% and top 5% percent based on risk score
3. Moves data from the last run to history tables for any patient whose risk score or risk level has changed since the previous run

To run the stored procedure:

1. From SSMS connect to the IRDS production database server
2. Select the Reach database, open a query panel and run the risk score stored procedure.

#### Patient Dashboard Stratification SQL script

The initial IRDS system presented all patients identified as "At Risk" (top 5% of the population) in the individual views of the IRDS dashboard. For each patient presented in the dashboard:

* Clinical support data (such as appointments, emergency contacts) is imported from CDW into IRDS
* A VLER direct message is sent to the users at a patient's primary VAMC

The VA determined that they still want to see aggregate totals at the facility and national views of the IRDS dashboard, but at the individual level only want to see a handful of randomly selected patients. Functionality was added to filter patients out of dashboard and also manage which patients to pull clinical data for and which to send VLER message for.

The stratification script, is run directly in SSMS against the production database. It:

1. Randomly selects patients per the current VA stratification specification to appear in the dashboard and flags them in the system
2. Adds a record to the VLER queue table for NODE JS VLER function to process

NOTE: If the stratification specifications change, per the VA, the script will need to be updated accordingly.

#### Clinical Support Import SSIS package

As mentioned in the previous section, for all patients presented in the IRDS dashboard, clinical support data, which appears in the dashboard widgets is imported from CDW.

To execute this package from the ETL Server remote desktop session go the file and execute the package as per the instruction in section 2.2.1.1.

#### Suicide Data Repository (SDR) Import SSIS package

This package pulls data from the production database on the SDR server and pulls Suicide Prevention Application Program (SPAN) data into the IRDS system, which is presented in the data entry widget of the IRDS dashboard.

To execute this package go the file and execute the package as per the instruction in section 2.2.1.1.

#### VLER Direct Message Node JS process

The VER Node JS Process reads the VLER queue from the Reach database, builds the emails notifying users of patients who have become at risk and delivers them accordingly.

To run the process

1. Remote desktop to the IRDS application server
2. Open Internet Information Services (iis)
3. From the connections panel select Sites
4. Click stop, then start.
5. Select the link in the “Browse Website” section on the right side of the window.

### Facility Surveillance Processes

There are two (2) SSIS packages that can, but do not necessarily need to be run during patient risk scoring processing. These packages support a dashboard widget that displays predicted attempt numbers per facility. These packages can be run per the data owner’s discretion.

#### Facility Surveillance Backend Process

This package pulls data from SPAN and utilizes an algorithm encapsulated in a series of R programs. The SSIS package passes data to the R programs, in an R environment on the IRDS production server. The R program output is imported into the Reach database to be displayed in the dashboard.

Because the package calls R programs on the IRDs database server, it is deployed on the server itself as opposed to the CDW ETL server. That is because CDW only allows a SQL connection to/from the IRDS servers.

Execute the package as per the instruction in section 2.2.1.1.

#### VAMC Year End Enrollment SSIS package

The facility surveillance algorithm, mentioned in the section above uses a data point which represents the relative size of each facility. To measure this value, the yearly enrollment sizes of each facility are queried from the CDW database and imported into the IRDS production database.

This package is run relative once a year, in January at the data owner’s discretion.

To execute this package go the file and execute the package as per the instruction in section 2.2.1.1.

## Security / Identity Management

The IRDS security architecture consists of components that perform authentication and authorization of VA. These components include the following:

* Network: Encrypted and secured communication between the Web application and the server over HTTPS
* Web application: Authenticated and authorized access to features
* CDW: Authenticated and authorized access to pull some basic data and upload data

The following attributes describe the IRDS architecture as related to security:

* The IRDS system resides in the AITC data center and consists of a web application, web services, and a database
* VA Users access authorized portions of the web application from VA facilities over VA VPN using the users network account credentials found within VA Active Directory.
* IRDS reads limited patient identification and demographics data from CDW and SDR, and writes results to the Reach DB
* IRDS integrates with CDW and SDR via SSIS entirely on the VA network

For more information on IRDS security, see the System Security Plan.

### Identity Management

Users are added, modified, or disabled through the Users table within the SQL Server Reach Database. In the Users table interface, the technical admin user has the ability to create users, assign their access (add them to program locations), and deactivate them. Adding a user consists of using the Userstable to fill in the new user’s username, user role, user state location, first name, last name, home facility, domain, and active/inactive status. Modifying a user consists of using the User table to modify values. Deactivating consists of changing the user’s status to *inactive*. For more information on user access roles and behaviors, see the User Guide and User Manual.

**Note:** Users are not deleted in IRDS; they are simply inactivated and they can be reactivated in the future as needed.

## Dashboard Management

The IRDS provides each user the capability to have their own experience with a personal dashboard. Users can modify the state of their dashboard by adding, moving, resizing, and deleting widgets. User’s dashboards are persisted within the IRDS Database. On occasion there may be issues with a user’s dashboard such as:

* A user receives a blank dashboard
* Configuration issue with dashboard
* Troubleshooting a dashboard due to data or other unknown reasons

Typically when a user experiences these issues with their dashboard, the dashboard can be deleted from the database according to the steps provided in Section 2.3.

## User Notifications

The user community will be notified of any scheduled changes via email distribution lists. It is recommended that separate mailing lists should be established for users, program administrators, and support staff.

## System Monitoring, Reporting & Tools

System monitoring should be performed and managed by the AITC staff assigned to the IRDS system. System monitoring tools should be established for operating system CPU, memory, disk space, and the IIS and SQL Server processes.

### Availability Monitoring

The IRDS availability monitoring requirements will be determined by the AITC staff assigned to the IRDS system. The main components that will be monitored are the web-based Dashboard application (IIS), database (SQL Server), and CDW connectivity.

| **Concern** | **Test** |
| --- | --- |
| Web-based Dashboard application | Application: OK |
| Database connectivity | Database: OK |
| CDW connectivity | CDW: OK |

Table 8: Application Status Checks

Loading the status screen checks the application status in general, as well as the database and CDW connectivity. In addition to automated monitoring, this screen can be checked manually to determine the status of the system after a deployment or patch, or during troubleshooting.

### Performance/Capacity Monitoring

IRDS performance and capacity management consists of two concepts: verifying system performance through page loads, and verifying capacity through disk and network analysis.

There are three stated Key Performance Indicators (KPI) for IRDS: initial page load <= 20s and subsequent page load <= 5s. These KPIs are summarized below:

| **Action** | **Threshold** | **Verification** |
| --- | --- | --- |
| Initial page load | 20 seconds | Manual |
| Subsequent page load | 5 seconds | Manual |

Table 9: Performance Thresholds

The responsibilities of performance and capacity monitoring belong to AITC, since the IRDS solution will be hosted within AITC servers. For more details on AITC’s procedures and monitoring tools please contact AITC.

### Critical Metrics

The critical metric for IRDS is whether 100 concurrent users can simultaneously use the system nationally. The downstream implication is a possible delay in identifying or seeking treatment for some Veterans. The critical metric is summarized below:

|  |  |  |  |
| --- | --- | --- | --- |
| **Metric** | **Threshold** | **Upstream implications** | **Downstream implications** |
| Concurrent users nationally | 100 | Delay or errors accessing the surveillance Dashboard application | Delayed identification of at-risk Veterans |

Table 10: Critical Metrics for IRDS

The system’s current or historical support for concurrent users/site can be assessed by exporting the log files to VA’s enterprise log analysis service. The logs have an industry standard structure that will be recognized without custom parsing by most commercial or open source log parsing tools. Adherence can be determined by comparing page requests times and error counts against concurrent logins.

## Routine Updates, Extracts and Purges

Updates, extracts, and purges are performed for IRDS per VA guidelines and as requested by program administrators. These activities are summarized below:

| **Activity** | **Periodicity** | **Responsible party** |
| --- | --- | --- |
| Updates | As needed | DBA |

Table 9: Routine Data Activities

Updates consist of inserting or updating data in the database can be performed as needed. Updates should be scripted with sufficient error handling and rollback logic to handle expected and unexpected errors during execution while protecting data integrity. Data changes require expertise in SQL and the IRDS schema (see project schema document). Updates should be performed by qualified DBAs as requested by IRDS program coordinators.

## Capacity Planning

Capacity planning should be performed by AITC in cooperation with IRDS program administrators. Currently, there is no schedule or requirements for capacity planning.

# Exception Handling

Runtime errors in IRDS are typically related to configuration, connectivity, or data issues. Errors related to connecting to the IRDS database, configuration, and bad or unmatched data can be resolved locally by the system administrator. Other kinds of errors, such as problems connecting to the CDW can be resolved through cooperating with external teams. The types of errors are summarized below:

| **Type** | **Examples** |
| --- | --- |
| Locally resolvable | Unmatched records, bad data, DB connectivity |
| Externally resolvable | Network or CDW issues |
| Unresolvable | Errors due to bugs |

Table 12: Types of Errors

**Note:** Some errors, such as those due to unidentified bugs, require application source code changes and cannot be changed by the system administrator.

## Routine Errors

Like most systems, IRDS may generate a small set of errors that may be considered routine in the sense that they have minimal impact on the user and do not compromise the operational state of the system. Most of the errors are transient in nature and only require the user to retry an operation. The following subsections describe these errors, their causes, and what, if any, response an operator needs to take.

While the occasional occurrence of these errors may be routine, observing a large number of an individual error over a short period of time is an indication of a more serious problem. In that case the error needs to be treated as an exceptional condition.

### Security Errors

Security errors in IRDS consist of authentication or authorization issues related to accounts. For example, if a staff user attempts to logon to IRDS with invalid credentials three contiguous times, the system will display an error message directing the user to contact the system administrator.

Likewise, if a staff user is already authenticated but attempts an unauthorized action in the application, the system will display an error message

### Time-outs

In IRDS, timeouts can occur between the client and the server, and between the server and CDW. Timeouts can be due to capacity issues with regard to the IRDS server, the CDW server, or the network fabric in between. It is expected that most timeouts will be due to capacity or contention issues caused by the browser communicating with the server over VA network, not on the server itself or between the server and CDW. When any part of the system times out, the application displays a user-friendly error message indicating that the user should talk to the clerk. The following table summarizes the types of possible timeouts:

| **Type** | **Incidence** | **Response** |
| --- | --- | --- |
| Timeout connecting to CDW | Unknown | Repeat attempt or file support ticket with the CDW group. |
| Timeout loading data to IRDS | Unknown | Try again, troubleshoot server, or file support ticket with NOSS or AITC |

Table 14: Possible IRDS Timeouts

Most timeouts will be transient in nature, and resolve after the network or server contention abates. However, timeouts can also be investigated and submitted to the appropriate support groups. Some timeouts between the server and client can be logged, timeouts on the server itself, and timeouts between the server and CDW are logged on the server. This allows the system administrator to investigate individual timeout issues as well as use system tools or external tools in order to investigate patterns of timeouts.

### Concurrency

Concurrent updates can lead to unpredictable errors in any system, including IRDS. However, due to the nature of IRDS, concurrency issues are very unlikely to occur. If they did occur, they would be related to very rare events like multiple staff attempting to update a Veteran’s outreach status in the application. In the case of concurrent updates to Veteran data, the system will note if one user is updating an old version of the data and prompt the user to view the updated record and possibly try again.

The table below summarizes the type of possible concurrency issues that could occur:

| **Activity** | **Incidence** | **Response** |
| --- | --- | --- |
| Simultaneous updates to Veteran outreach status in the Dashboard | Very rare | System will catch and log error, then prompt user to (optionally) try again. |

Table 15: Possible Concurrency Issues

## Significant Errors

Significant errors can be defined as errors or conditions that affect the system’s stability, availability, performance, or otherwise make the system unavailable to its user base. The following subsections contain information to aid administrators, operators, and other support personnel in the resolution of errors, conditions, or other issues.

### Application Error Logs

IRDS logs are currently limited to SQL Server Database logs.

Logging for each component is configurable at the component level. The log configuration files are stored in the following folders:

Logging is configured by the system administrator. Sensible defaults are supplied along with the application, allowing adequate log coverage for troubleshooting without affecting performance or taking up excessive disk space. The following table outlines the key logging attributes:

Querying and analyzing the log files is simple because they are text files and use the industry-standard log conventions (INFO, WARN, ERROR, etc.).

### Application Error Codes and Descriptions

IRDS does not currently use error codes; rather, it defines custom Java Script Object Notation (JSON) object that can be used for exception handling. This JSON object can be reused across a family of issues. The table below describes the existing custom exception types and descriptions:

| **Exception Type** | **Description** |
| --- | --- |
| Bad Password Exception | This password is not correct. |
| Bad UserID Exception | User not registered/Invalid Username |
| Inactive User Exception | Account is disabled due to inactivity, Contact system admin |
| Locked User Exception | Account locked, Please contact system admin |
| Max Sessions Reached Exception | Max number of sessions reached, Please log out from your active sessions or wait for 30 seconds and try again |

Table 18: Existing Custom Application Types and Descriptions

### Infrastructure Errors

IRDS relies on various infrastructure components and must handle temporary failures in those components when they occur.

#### Database

IRDS can experience errors connecting to the database or performing data operations. Database connection errors can be found in the logs by querying for “connection” and orphaned connections can be queried and forced close via SQL Server commands. For more information on querying and force-closing orphaned connections, see the SQL Server online manual: <https://support.microsoft.com/en-us/kb/137983>.

The application can experience errors performing data operations as well. This includes errors querying, inserting, updating, or deleting data. When these types of database errors occur, the application will catch the exception and log it. If the error is something that the user can fix by trying again, the application will display a message to the user; otherwise, the application will handle the error itself and may direct the user to a user-friendly error page based on the severity of the error.

#### Web Server/Application Server

Node JS automatically logs all errors to the *stderr* and *stdout* files, although the system administrator can configure the logging per VA guidelines. However, leveraging IIS allows for easier configurability and additional options for logging capabilities. Errors are denoted in the logs by severity (e.g., “SEVERE”).

By default, IIS uses IIS Logging File Format; however, for the system administrator can easily configure IIS to use W3C, NCSA, or custom logging formats instead as per VA conventions. For more information on IIS logging, see the online manual ([www.iis.net/learn/manage/provisioning-and-managing-iis/configure-logging-in-iis](http://www.iis.net/learn/manage/provisioning-and-managing-iis/configure-logging-in-iis) ).

The IRDS web application is currently configured to do its logging through IIS. The system administrator can configure application-specific behavior in the IIS log configuration utility.

#### Network

IRDS can suffer from errors due to network conditions between the client and the server, or between the server and CDW. If there are network problems during the initial loading of a page, the client may display built in error messages (e.g., HTTP 404). On the other hand, if there are errors transmitting data in the background, the client JavaScript will attempt to retry the operation before failing with a user-friendly error message.

For network issues between the server and CDW, if the application can catch and retry the operation, it will. For network errors beyond the application’s grasp, the server will fail and log the operation and redirect the user to a user-friendly error page. The error page typically instructs the user to see the system administrator.

#### Authentication & Authorization

All authentication and authorization errors are caught by the application and logged.

For IRDS authentication errors, the system will prompt the user a total of three times and then lock the user account and provide them with a message to see the system administrator. IRDS authorization errors should be rare, but if they occur, the user will be notified to contact their system administrator. The administrator can adjust the user’s settings as needed.

## Dependent System(s)

IRDS is dependent upon CDW for retrieving Veteran Personally Identifiable Information (PII)/Protected Health Information (PHI) data. IRDS is also dependent on VLER Direct to submit secure messages to outreach personnel regarding high risk veterans. For persistent failures connecting to CDW or VLER Direct, there’s nothing that can be done to resolve access or connectivity issues within IRDS; all other errors must be resolved in cooperating with CDW or VLER Direct support.

## Troubleshooting

Troubleshooting IRDS issues consists of checking the logs and tweaking configuration settings. Most application behavior cannot be adjusted without modifying code. The following table summarizes the types of errors and resolution procedures likely to occur in IRDS:

| **Type** | **Procedure** |
| --- | --- |
| Errors | Check logs and report issue. |
| Database connectivity issues | Check status page, application and database logs and connection string. Troubleshoot using SQL Server Management Studio client. |
| Other database issues | Check logs and report issue. |
| CDW connectivity issues | Check logs, report issue to appropriate help desk. |

Table 19: Troubleshooting IRDS

The first step in most cases is to check the system status page. The process of authenticating and viewing the status page will give you some information about the system stability, because this process exercises the application, database, and CDW. The inability to authenticate errors reported on the status page allows the system administrator to narrow his/her focus.

The next step is to check the logs. The logging level can be temporarily dialed up in each logging configuration file (see 3.2.1) to support DEBUG-level messaging as needed. The logs will display detailed information about the type of problem that is occurring, and can be tailed and searched. If the application is operational in general, failing actions can be tested in the application and checked in the logs.

If the application cannot connect to the database, the system administrator can check whether the *MSSQLSERVER* process is running, check the SQL Server logs, and test the connection to the database using various parameters using the SQL Server Management Studio client (application interface). The SQL Server Management Studio client can also be used to query and modify data or state as needed in order to resolve the issue.

If the problem is with CDW, the connection information can be changed or confirmed with CDW support technicians. Most CDW issues will require cooperation with CDW support.

If the problem is with VLER Direct, the connection information can be changed or confirmed with VLER Direct support technicians. Most VLER Direct issues will require cooperation with VLER Direct support.

Finally, if the issue lies within the application itself, such as a bug or the inability to deal with an unforeseen issue in the production environment, the application source code can be modified as needed to resolve the issue.

The following tables provide a detailed listing of error conditions and resolution actions.

|  |  |  |
| --- | --- | --- |
| **Category** | **Description** | **Actions** |
| Verify IRDS Account in the Login Page | User is presented with “Account is locked. Please contact System Admin” | User’s account is locked in IRDS. User has entered the wrong username and password more than 3 times. User should seek assistance with VA Active Directory admin to ensure their username and password is correct or have it be reset to something else. Then the user needs to contact the IRDS System Administrator to resolve the locked account issue. |
| Verify IRDS Account in the Login Page | User is presented with: “Max number of sessions reached, Please log out from your active sessions or wait for 30 secs and try again” | Users login attempt is blocked due to max user sessions already opened. User must close active sessions by logging out successfully and try again or wait 30 secs if browser window was closed prior to successful logout. If problem still persists please contact the IRDS System Administrator to resolve the issue. |
| Verify IRDS Account in the Login Page | User is presented with: “Account is disabled due to inactivity, Contact system admin.” | User has entered valid account credentials however it has been more than 30 days since last login attempt. After 30 days of no login activity, user account will be deactivated and will need to contact IRDS System Administrator to resolve the issue. |

Table 20: Errors and descriptions

| **Category** | **Actions** |
| --- | --- |
| Unable to Connect | The server cannot be connected to from the terminal VA staff is using. The network used by the resource should be checked to make sure it has a network connection (e.g. navigate to a page within the VA’s network). If the device can navigate to another VA site then the server should be checked to make sure it is still running. If it is, then IT should be contacted to ensure that the network hasn’t become fragmented. |
| Database Connection Issues | As a system administrator, log into the web server and ensure the web server can communicate with the database, ensure the database is running by checking the Windows service list, and ensure the database account the web site is using is able to be used to log into SQL Server |
| CDW Connection Issue | As a system administrator, log into the web server and ensure the web server can reach the CDW server. Contact the CDW administrator to verify existing connection settings are still valid. |
| VLER Direct Connection Issue | As a system administrator, log into the web server and ensure the web server can reach the VLER Direct server. Contact the VLER Direct administrator to verify existing connection settings are still valid. |

Table 21: Actions for common errors

## System Recovery

The following subsections define the process and procedures necessary to restore the system to a fully operational state after a service interruption. Each of the subsections starts at a specific system state and ends up with a fully operational system. Reference the IRDS Installation Guide for additional description of processes including working with AITC server owners for system recovery.

### Restart after Non-Scheduled System Interruption

If the system crashes or is brought down, it can be simply restarted by restarting the database and IIS server processes and then viewing the application status page. The two processes, which are covered earlier in this document, are run as Windows services and can be started from the Services snap-in. Once the services start, the system administrator can logon to the application and view the status page in order to verify connectivity. The full steps are as follows:

1. Ensure the SQL Server and IIS services are running.
2. Load the application home page
3. View the status page to ensure the application can connect to CDW

If the Windows services do not start properly, the Event Viewer and the log files for each service can be checked for errors. Failures in the services are unlikely to happen, however, if the operating system itself is healthy.

### Restart after Database Restore

The system can be restarted after restoring from a database backup by simply accessing the application. If the application server was taking offline in order to prevent access to the database during the restore, the application server should be restarted as well before utilizing the application.

### Back Out Procedures

The upgrade back out procedure consists of notifying the service desk, taking the application offline such that users see a “down for maintenance” page, performing the back out steps (restoring the database, redeploying the old version of the application, etc.), checking the application locally, restoring service, and then notifying the help desk that the maintenance is over. Restoring previous version from archive directory.

# Operations & Maintenance System Support

An understanding of how IRDS is supported by various organizations within the VA is important to operators and administrators of the system. If you are unable to resolve an issue, then it is necessary to understand how to obtain support through the Office of Information and Technology (OI&T) system support organizations. The following sections describe the support structure and provide procedures on how to obtain support.

## Support Structure

This section describes the systems support structure as seen from the perspective of operations personnel. The first section defines the support hierarchy through which a support request may navigate. The second section defines the responsibilities for each level of support.

### Support Hierarchy

During the pilot phase of IRDS, the Perceptive Reach development team will provide all Tier 1 and Tier 2 support. Perceptive Reach support staff will contact AITC as necessary.

## Support Procedures

The IRDS support procedures will consist of triage, troubleshooting, and change management.

1. Defect and change requests triaged by Program Administrator
2. Troubleshooting by Healthcare System Technical Administrator
3. Change management performed by application developers as authorized by the application’s Change Control Board

Assigned System Administrators from the Perceptive Reach development team will provide primary support for all of these tasks. As needed, these resources will contact other VA teams and organizations, such as AITC, as required during the execution of these activities. Enhancements, bug fixes, etc. as identified in step three listed above will be completed in accordance with Agile / Scrum principles within the application’s regular software development cycles.

# Acronyms & Abbreviations

|  |  |
| --- | --- |
| **Acronym** | **Term** |
| AITC | Austin Information Technology Center |
| CDW | Corporate Data Warehouse |
| DB | Database |
| IRDS | Integrated Reach Database System |
| KPI | Key Performance Indicator |
| JSON | Java Script Object Notation |
| OI&T | Office of Information and Technology |
| PHI | Protected Health Information |
| PII | Personally Identifiable Information |
| SAM | System Administration Manual |
| SDR | Suicide Data Repository |
| SPC | Suicide Prevention Coordinator |
| SQL | Structured Query Language |
| SSIS | SQL Server Integration Services |
| SSMS | SQL Server Management Studio |
| SPAN | Suicide Prevention Applications Network |
| VA | Department of Veterans Affairs |
| VBA | Veteran Benefits Administration |
| VHA | Veterans Health Administration |
| VLER | Virtual Lifetime Electronic Record |

Table 22: Acronyms & Abbreviations

# 