Software Development Plan for the Regional Integrated Corridor Management System

Version: 1.0

Date: July 6, 2018

Approval date: July 5, 2018

DOCUMENT CONTROL PANEL		
File path and name:	https://fldot.sharepoint.com/sites/FDOT-EXT-TEO/D5TSMO/DataInitiatives/ICMS/Shared Documents/03 Project Management/Software Development Plan/R-ICMS-SDP-1.0.docx	
Version Number:	1.0	
	Name	**Date**
Created By:	Robert Heller, SwRI	May 3, 2018
	Angela Bos, SwRI	May 21, 2018
	Clay Weston, SwRI	May 21, 2018
Reviewed By:	David Vickers, SwRI	May 24, 2018
	Josh Johnson, SwRI	May 24, 2018
	Clay Packard, FDOT/VHB	June 1, 2018
Modified By:	Robert Heller, SwRI	July 5, 2018
Approved By:	Clay Packard, FDOT / VHB	July 5, 2018

DOCUMENT REVISION HISTORY			
Version Number	**Approval Date**	**Description of Change(s)**	**Created/ Modified By**
1.0	July 5, 2018	Initial Release	SwRI Team

Table of Contents

List of Tables

List of Figures

List of Acronyms and Abbreviations

API ..... Application Program Interface
ATP ..... Acceptance Test Procedure
BE521 ..... R-ICMS Contract
CAP ..... Corrective Action Plan
CM ..... Configuration Management
COTS ..... Commercial Off the Shelf
CSAR ..... Configuration Status Accounting Report
DAG ..... Directed Acyclic Graphs
DFE ..... Data Fusion Engine
DSS ..... Decision Support System
ETL ..... Extract Transform Load
FDOT ..... Florida Department of Transportation
IEN ..... Information Exchange Network
ITN ..... Invitation to Negotiate
IV&V ..... Independent Verification and Validation
ITS ..... Intelligent Transportation Systems
ME ..... Modeling Engine
MSR ..... Monthly Status Report
PSEMP ..... Project System Engineering Management Plan
R-ICMS ..... Regional – Integrated Corridor Management System
RTVM ..... Requirements Traceability Verification Matrix
SAT ..... System Acceptance Test
SCM ..... Software Configuration Management
SDLC ..... Software Development Lifecycle
SDP ..... Software Development Plan
SQA ..... Software Quality Assurance
SQL ..... Structured Query Language
SRS ..... Software Requirements Specification
SwRI ..... Southwest Research Institute
UI ..... User Interface
VDD ..... Version Description Document

1 Overview

The Software Development Plan (SDP) establishes the software development approach, methodologies, tools, and procedures Southwest Research Institute® (SwRI®) will use during the analysis, design, development, testing, integration, deployment, and maintenance of the software for each Florida Department of Transportation (FDOT) Intelligent Transportation Systems (ITS) project. This SDP is a dynamic document and shall be updated on a periodic basis to reflect organizational changes, lessons learned, new tools, and advances in methodologies. The SDP will not be directly applicable to the FDOT ITS engineer, since typical FDOT activities do not usually include software development. However, the SDP should be a requirement for the project subcontractors responsible for developing and submitting the SDP document for a software development effort.

1.1 Scope

The SDP provides the means to coordinate schedules, control resources, initiate actions, and monitor progress of the development effort. The purpose of the document is to provide a detailed plan for the use of resources, methodologies, and techniques that provide for the development of all software that comprise the product line.

1.2 Identification

Project Name: *Central Florida Regional Integrated Corridor Management System*.
Agreement Number: BE521
Financial Project Identification: 436328-1-82-01
Federal Aid Project Number: *Not Applicable.*

1.3 System Overview

The R-ICMS will consist of, but not be limited to; commercial off-the-shelf (COTS) modeling software (provided by the DEPARTMENT), a custom built Decision Support System (DSS), a custom built Information Exchange Network (IEN) subsystem that includes dashboards and other user interfaces to the system, and a Data Fusion Environment (DFE) to host data sources for both the R-ICMS and other external users and applications.

1.4 Relationship to Other Plans

This SDP refers to other documents that will be produced during the course of the project. These include:

R-ICMS-ACTLog	The ACTLog is a spreadsheet that contains multiple sheets relevant to this SDP including: Action Item List, Change Management Log, Communications Log, Risk Register, etc.

R- ICMS-PSEMP	The Project Systems Engineering Management Plan (PSEMP) documents certain processes and procedures for the technical management, procurement, installation, and acceptance of the project. There is overlap between this SDP and the PSEMP.
R-ICMS-RTMV	The Project Requirements Traceability Verification Matrix shows bi-directional traceability from user needs statements to Acceptance Test Cases.
R-ICMS-REQ	The Software Requirements Specification provides a list of all requirements the system will satisfy with final delivery.
R-ICMS-SCHED	The Project Schedule that shows activities through the life of the project.

1.5 Applicable Documents

The following documents, of the exact issue shown, form a part of this document to the extent specified herein. In the event of a conflict between the contents of the documents referenced herein and the contents of this document, this document shall be considered the superseding document.

Standard Written Agreement, Agreement Number BE521	*FDOT District 5 Procurement. A copy is maintained on the Project SharePoint Site.*
ITN-DOT-16-17-5004-ICMS	*FDOT District 5 Procurement. A copy is maintained on the Project SharePoint Site.*
Systems Engineering and ITS Architecture (Topic No 750-040-003)	http://www.dot.state.fl.us/proceduraldocuments/procedures.shtm

2 Overview of Software Development Planning

This section establishes the context for the planning described in later sections. Include an overview of:

- Requirements of the system and software
- Project documentation
- Identification of the Software Development Life Cycle (SDLC)
- Software development team training

2.1 Requirements and Development

System and software requirements will be documented in the requirements database. Software requirements will be derived from the system requirements and allocated to computer software modules.

The FDOT developed requirements for Regional Integrated Corridor Management System (R-ICMS) and documented those requirements in Exhibit C of the Invitation to Negotiate (ITN). Negotiations reviewed and pruned those requirements, resulting in the requirements that appear in the contract, BE521. The major changes from the requirements of the ITN to those of the contract are reduction of the number of data sources and providing an interface to a commercial off the shelf (COTS) traffic Modeling Engine (ME) acquired by FDOT under separate contract instead of requiring this project to acquire the ME.

2.2 Project Documentation

SwRI will produce project documentation using the Microsoft Office Suite of tools.

- Word will be used to document project plans including
    - PSEMP
    - Monthly Status Reports
    - Software Development Plans and Coding Standards
    - Software Preliminary and Detailed Designs
    - Software Test Plans, Procedures and Reports
    - Hot Wash-up Minutes
    - Meeting Agendas and Minutes
    - Training Plans, Training Materials, Training Schedules
    - Other documents
- Use Excel and the ASTLog spreadsheet to document Risks, Action Items, Issues, etc.
- Use Excel to document and exchange the system and software requirements
- Use PowerPoint to prepare all presentations for meetings, training and technical interchanges
- Use Schedule to prepare and maintain the project schedule.

SwRI will deliver the prepared documentation via email to the FDOT Project Management team (e.g. Tushar Patel, Jeremy Dilmore and Clay Packard) and the SwRI Project mailing address (Project23368@SwRI.org). Concurrently the prepared documentation will be stored on the FDOT hosted project SharePoint site. SwRI maintains a Configuration Managed repository of all delivered documentation.

2.3 System Development Life Cycle

SwRI will use an Iterative SDLC for the development consisting for four iterations as illustrated in Figure 1.

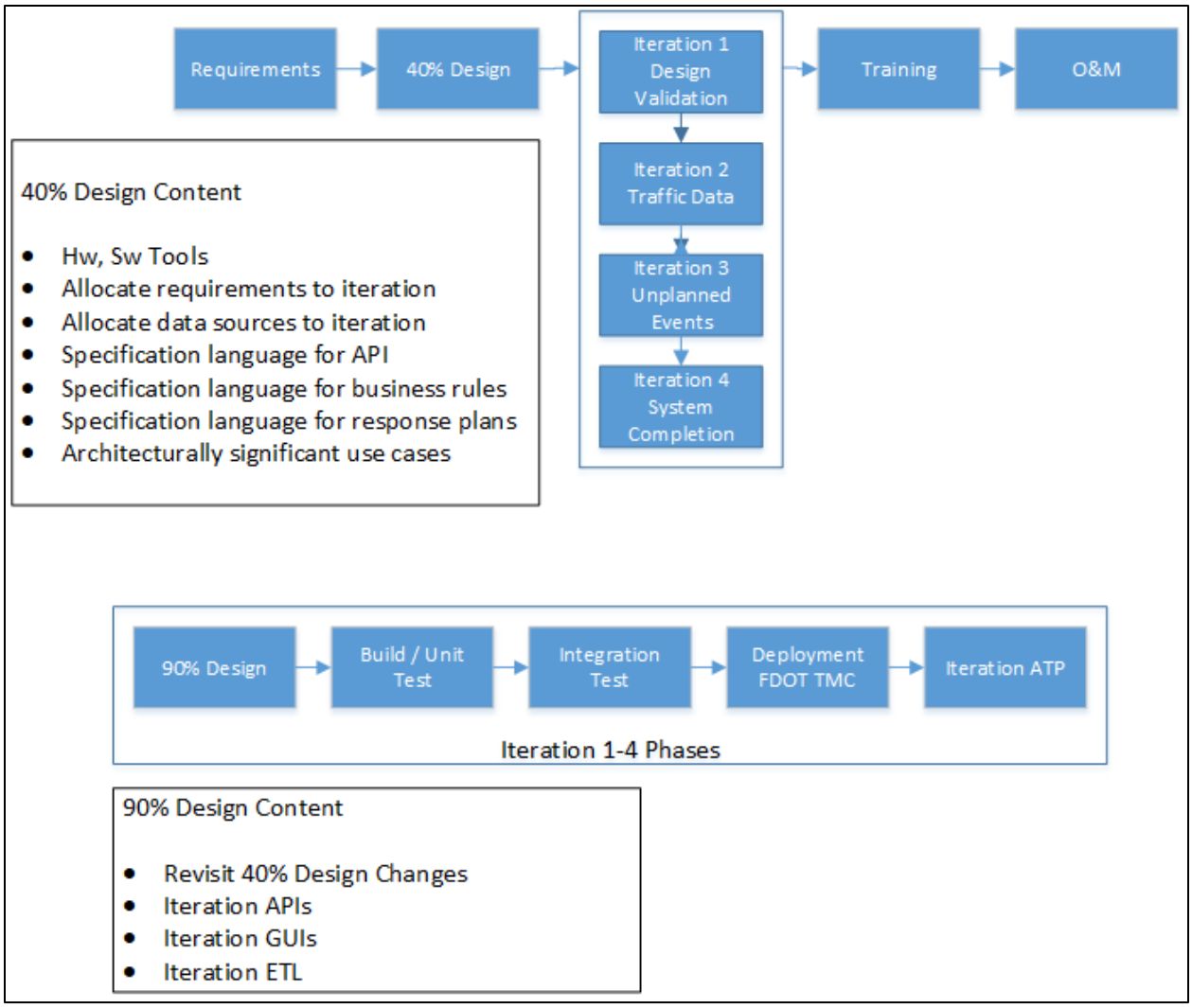

Figure 1 -- R-ICMS SDLC

2.4 Schedules and Resources

The project schedule is maintained as a Microsoft Project file, it may be found on the FDOT hosted Project SharePoint site.

2.5 Training Requirements

The SwRI development team has necessary training to conduct this project. If the team identifies additional training necessary, it will be the responsibility of the team members to provide that training. The SwRI project team has been formed to include experience with the following software technologies.

- Microsoft Visual Studio
- Microsoft SQL Server Relational Data Base Management System
- Hadoop big data environment and the Cloudera support tools
- Elastic Search for storage and search support of unstructured data
- Geographic Information Systems especially the ESRI toolset and Open Source tools
- C#, Node JS, Angular/ SyncFusion UI Controls, jQuery, Bootstrap, HTML5/CSS, JavaScript
- Java, Eclipse Studio, IntelliJ IDEA
- Crystal Designer

3 General Software Development Activities

This section describes general software development activities as they pertain to this project.

3.1 Development Process

SwRI will follow an iterative SDLC as depicted in Figure 1. The following paragraphs describe in more detail the content of the iterations. Each iteration is based on processing additional data sources in that the Data Fusion Engine (DFE) will ingest the new data sources, the Information Exchange Network (IEN) will display the new data sources and the Decision Support System (DSS) will process the new data sources. At the conclusion of each iteration, the following deliverables will be provided: source code and other source artifacts, executables and other implementation artifacts, and a version description document (VDD), interface control document (ICD), and system user manual (SUM) updated with content related to the delivered prototype. At the conclusion of the fourth iteration, the system will ingest, display and process all contractually required data sources and the completed versions of source artifacts, executable artifacts, and documentation artifacts will be delivered for the entire R-ICMS system.

3.1.1 Iteration 1: Design Validation

The functionality to be implemented during the Design Validation includes components and capabilities that are architecturally significant or that illustrate the use of the selected tools. The validation capabilities to be prototyped for the validation phase are:

DFE Design Validation

Definition: DFE will receive static data for transit sources, intersection data, RCI data and schools related data, store that data, and make the data available via a defined Application Program Interface (API). Additionally, SwRI shall propose one (1) dynamic data source to be implemented as part of Iteration 1; this data source will be subject to FDOT approval during requirements analysis.

Demonstrate: Show the receipt of data through data logs, the storing of the data in the logs and the availability of the data through the API accessed by a test jig.

Deliverable: The source code that implements the extract-transform-load (ETL) and the API, the test jig, the API definition (ICD) and the data storage schema.

IEN Design Validation

Definition: The IEN will display the static data listed in the DFE milestone above on a graphical user interface and the one (1) TBD dynamic data source.

Demonstrate: Display of the static data.

Deliverable: The display code and rudimentary user manual pages describing the use of the IEN to display static data.

DSS Design Validation

Definition: The Highway Capacity Software (HCS) backend for the SOT will be encapsulated and programmatically used to produce a set of optimum signal timing plans for a single-intersection corridor.

Demonstrate: The production of a file containing an optimized set of signal timing plans for the demonstration intersection.

Deliverable: The code that encapsulates the HCS backend, a rudimentary user manual for the manual initiation case for optimization of a single series of intersections (aka known as an intersection corridor).

3.1.2 Iteration 2: Traffic Data

Iteration 2 concentrates on the retrieval, storage and display of traffic data.

DFE Milestone

Definition: DFE will receive traffic data, store that data, and make the data available via a defined API.

Demonstrate: Show the receipt of data through data logs, the storing of the data in the logs and the availability of the data through the API accessed by a test jig.

Deliverable: The code that implements the ETL and the API, the test jig, the API definition (ICD) and the data storage schema.

IEN Milestone

Definition: The IEN will display the traffic data on a graphical user interface and implement other traffic oriented displays per the Prime Contract.

Demonstrate: Display of the traffic data.

Deliverable: The display code and rudimentary user manual pages describing the use of the IEN to display traffic data.

DSS Milestone

Definition: Bulk load rules into the DSS rules engine, bulk load response plan elements and response plans; develop method for wrapping HCL for use in SOT.

Demonstrate: Demonstrate ability to load rules, response plan elements and response plans through simple stub that dumps the rules, etc.

Deliverable: Bulk rule loader code, ICD for bulk rule loaders, bulk response plan loader, ICD for bulk response plan loader, schema for rules and response plans. Manual pages for SOT and bulk loaders.

3.1.3 Iteration 3: Unplanned Events

Iteration 3 concentrates on the retrieval, storage and display of event data from SunGuide and other sources.

DFE Milestone

Definition: Data is being extracted from event sources and is available from the DFE through the API.

Demonstrate: Show the receipt of data through data logs, the storing of the data in the logs and the availability of the data through the API accessed by a test jig.

Deliverable: ETL, “database” schema, API definition, API implementation.

IEN Milestone

Definition: Event data in the DFE can be displayed by the IEN and data can be entered into the DFE from the IEN.

Demonstrate: The IEN can display data from SunGuide, entry of events.

Deliverable: The display code and rudimentary user manual pages describing the use of the updated IEN.

DSS Milestone,

Definition: DSS will detect events, using rules to select response plans and invoking PRE (modeling engine or its stub). Manual SOT including signing of optimized signal timing plans.

Demonstrate: The rules engine invoking the modelling engine to process / model / predict.

Deliverable: Code; Modeling Engine ICD.

3.1.4 Iteration 4: Planned Events

Iteration 4 completes the DFE, IEN and DSS and concentrates on the retrieval, storage and display of planned events and signal optimization.

DFE Milestone

Definition: DFE will incorporate data from all remaining data sources.

Demonstrate: Show the receipt of data through data logs, the storing of the data in the logs and the availability of the data through the API accessed by a test jig.

Deliverable: ETL, API and associated source code and applicable documentation.

IEN Milestone

Definition: IEN will include screens for remaining functionality. Notably, these include support for planned events, response plan (tracking, evaluation and analysis), event workflow, etc.

Demonstrate: Demonstrate the functionality of the IEN to input, display and edit planned event Iterations, interface with the DSS in presentation, rating and selection of response plans and sending of the response plan elements to SunGuide.

Deliverable: Final pre-testing source code drop and associated documentation.

DSS Milestone

Definition: SOT will be fully automated.

Demonstrate: Demonstrate the ability of SOT to make signal timing modifications running in an automated manner.

Deliverable: DSS source code drop including SOT final code delivery.

3.2 Development Methods

The SwRI team will use industry standard software development methods appropriate to the various development tasks. In support of these methods, manual and automated tools and procedures will be used to include those listed below and others as needed.

Software development will follow an iterative release cycle, with four planned major releases, using a Kanban agile methodology with open communication and full transparency of work progress. A release management team will migrate issues developed from customer requirements and reported bugs into the work queue for a release. Developers will perform work from the queue, updating issue statuses as readiness of features and fixes progress through development, integration, testing, and production release. This work will be organized using the Atlassian collaboration toolchain:

- Jira for project and issue tracking
- Confluence for document collaboration
- HipChat for group communication
- Bitbucket for Git code management and code review processes and documentation
- Sourcetree for visual code management and merges
- Bamboo for integration and release management, using automated builds and deployment

The following software development practices and paradigms have been identified as having significant impact to the design and implementation of this project:

- Object oriented methodology with modular, re-usable components forming extensible, flexible application frameworks
- Event driven systems using scalable service-based architectures and real-time stream processing engines, such as Kafka and Spark, to develop directed acyclic graphs (DAG) of complex event processing
- Service oriented architecture
- MVC, MVVM, and Component Based User Interface (UI) design patterns
- Role based access control (RBAC), implemented using Active Directory and LDAP, implementing principles of least privilege
- Test driven development and automated testing using Selenium, Bamboo Server, and various Unit Test frameworks

3.3 Product Standards

SwRI will utilize the Atlassian products (Jira, RMsis) to represent requirements, test cases, test procedures and test results. Standard work flows will be configured, within RMsis and Jira, to ensure traceability from requirements to development tasks, to test cases. Standard design diagrams (UML, Use Case Diagrams, Class Diagrams, Sequence Diagrams, Systems/Component Diagrams) will be used to define the architecture of the underlying systems. Well-documented coding standards will be utilized and automated tools (Resharper, StyleCop) will ensure code compliance.

3.4 Reusable Products

SwRI will ensure that the system follows an extensible modular design. SwRI has identified the following areas of the system where user needs may require new or different functionality and will design the system to support changing these features with minimal impact on the overall architecture.

- Add new UI functionality
- Add new data sources
- Add new types of business rules
- Add new role-based access controls
- Flexible back end services (databases, modeling engines, etc.)

Additional opportunities for code/component reuse will be identified during the project lifecycle and similar flexible/hierarchical frameworks will be designed to accommodate them.

3.5 Computer Hardware Resource Utilization

SwRI developed a proposed hardware configuration and usage allocation during negotiations as shown in Table 1. There resources will be provided by FDOT and will be made available by acceptance test of the second iteration.

Table 1 -- Proposed Hardware Configuration & Utilization

Purpose	Server Count	Virtual Physical	CPUs	Cores/ CPU	RAM (GB)	Storage	OS
Virtual machines for Web servers and API servers (IEN and DFE)	6	V		2	128	2 x 1TB	Windows
Virtual machines for Web servers and API servers (IEN and DFE)	8	V		2	128	2 x 1TB	Linux
Kafka Cluster	3	P	2	8	128	2 x 1TB RAID 1 4 x 2TB RAID 1	Linux
Cloudera Manager	1	P	2	8	128	2 x 1TB RAID 1 2 x 1TB RAID 1 2 x 1TB Each	Linux
Hadoop Master nodes	2	P	2	8	128	2 x 1TB RAID 1 2 x 1TB RAID 1 2 x 1TB Each	Linux
Cloudera Data Nodes	9	P	2	8	128	2 x 1TB RAID 1 6 x 2TB Each	Linux
Edge nodes for client sessions / users	3	P	2	8	128	2 x 1TB RAID 1 1 x 1TB Each	Linux
Edge nodes for ETL	3	P	2	8	256	2 x 1TB RAID 1 6 x 2TB Each	Linux
Elastic search Master nodes	2	P	2	8	128	2 x 1TB RAID 1 1 x 1TB Each	Linux
Elastic search worker nodes	3	P	2	8	128	2 x 1TB RAID 1 2 x 8TB RAID 1	Linux
Servers for hosting NoSQL database (e.g. MongoDB)	3	P	2	8	128	2 x 1TB RAID 1 6 x 2TB Each	Linux
Virtual Hosts	2	P	4	16	256	10 x 2TB (RAID 10)	Windows

4 Detailed Software Development Activities

This section provides the plans for performing detailed software development activities, including the approach (i.e., methods, procedures, and tools).

4.1 Project Planning and Oversight

SwRI will develop and deliver a schedule in Microsoft Project format showing the project tasks, deliverables, scheduled meetings and reviews. The schedule will show predecessor relationships and the critical path. Additionally, SwRI will provide FDOT insight on a periodic basis into the project progress so it may perform necessary project oversight.

- SwRI will provide the FDOT with Monthly Status Reports (MSR) describing the progress status during the reporting month. The MSR will document all tasks completed, tasks underway and tasks to be started during the next reporting month.
- An attachment to the MSR will be a printed output of the ASTLog status spreadsheet that will document the Risks, Action Items, and Issues.
- SwRI will participated in Monthly Status Meetings with the FDOT; these meetings will be held via screen share; quarterly these meetings will be held in FDOT District 5 facilities. The agenda for these meetings will include
    - Review of new or changed risks
    - Action item status
    - Issues list
    - Status of deliverables
    - Status of milestones and invoices

4.2 Establishing a Software Development Environment

SwRI and EPIC will provide its own locally hosted Software Development Environments. The software development environments will consist of physical and virtual servers to host the necessary software development tools including:

- Linux Hosted Tools
    - Cloudera Hadoop Enterprise
    - Cloudera Kafka Distribution
    - MongoDB Enterprise
    - Elastic Search Enterprise
- Windows Server Hosted Tools
    - ArcGIS
    - SQL Server
    - IIS
- Atlassian Toolchain
    - Bitbucket Source Code Control System for Git

    - Jira Issue Tracking Tool
    - RMSIS Requirements Management Tool
    - Bamboo Server
    - HipChat
- Integrated Development Environments (IDE)
    - Windows Visual Studio
    - Eclipse Studio
    - IntelliJ (IDEA, PyCharm)

4.3 System Requirements Analysis

FDOT District 5 Traffic Operations developed a Concept of Operations and a High Level Functional Requirements document which is published in the procurement document ITN-DOT-16-17-5004-ICMS. The former is captured in Exhibit B Scope of Services, the latter are captured in Exhibit B System/Subsystem Minimum Technical Requirements for the Central Florida R-ICMS.

4.3.1 Requirements Walkthrough

SwRI will host a weeklong on-site requirements walkthrough meeting. The purpose of the Requirements Walkthrough meeting is to identify those requirements that are found to be:

- Duplicates and remove or combine them,
- Compound and need to be broken into smaller single function requirements (children),
- Outdated and need to be removed,
- Design requirements and not functional requirements which need to be removed.

At the conclusion of the Requirements Walkthrough, SwRI will deliver to FDOT a list of the agreed and approved changes to the requirements originally published in the ITN.

4.3.2 Requirements Specification

SwRI will utilize the output of the Requirements Walkthrough to produce the Requirements Specification. By agreement, the Requirements Specification will consist of a cover page and an Excel Spreadsheet consisting of the updated requirements.

- SwRI will apply the approved changes from the Requirements Walkthrough as enumerated in the previous section.
- SwRI will analyze all requirements to ensure they are unambiguous and testable. In those cases where requirements are found not to be, SwRI will either
    - Sketch simple use cases to derive and understanding of the requirement and propose updated language for the requirement
    - Leave the requirement as an open question to be answered by a Stakeholder.

- SwRI will update the requirements Excel Spreadsheet, showing how the new requirements were derived, modified, or other changed.
- SwRI will deliver the updated requirements spreadsheet to the FDOT for approval.

4.3.3 Requirements Database

SwRI will populate the RMSIS requirements management tool of the Atlassian toolset with the approved requirements. The RMSIS tool will be hosted at SwRI and the UI exposed to the Internet with secured access granted to other members of the SwRI team and identified R-ICMS stakeholders. SwRI will use the RMSIS toolset through the remainder of the project to track requirements changes and traceability.

4.3.4 Requirements Traceability and Verification Matrix (RTVM)

SwRI will populate appropriate fields of the RMSIS requirements management tool to trace requirements from User Needs, Business Needs, through all stages of Requirements Analysis, Design, Implementation and Testing.

- Requirement Analysis Phase: create the RTVM as part of the System Requirements Specification (SRS). The SRS will consist of the requirements from Exhibit C as modified or augmented by the efforts during and following the Requirements Walkthrough.
- 40% Design Phase: allocate requirements to specific architectural design artifacts. This assures that all requirements have been considered during the physical architecture of the system.
- 90% Design Phase: further allocate requirements to specific software modules or other software artifacts e.g. database tables, interfaces, etc.
- Unit Test: add unit test case identifiers to verify that allocated requirements were tested
- Integration Test: add integration test case identifiers to verify that allocated requirements were tested during the integration test phase.
- Acceptance Test: add acceptance test case identifiers to verify that the system satisfies system requirements.

SwRI will configure the RMSIS tool to produce on-line viewable traceability and on-demand produce printable RTVM reports.

4.4 Software Implementation and Test/Code and Unit Test

SwRI will develop fully independent unit tests that completely encapsulate the setup, execution, and validation of single logical units of functionality where appropriate. Unit Test methods will vary by system component and may include both manual and automated implementation. Manual testing procedures will be documented in the associated Jira issue tracking the work. Automated tests will be developed using standard tools for the programming language or software development kit, such as:

- Java: JUnit, TestNG, Maven
- Python: unittest, pytest, nose
- Scala: ScalaTest
- .NET: Microsoft Unit Test Framework
- Angular/JS: Karma, Jasmine, angular-mocks
- UI: Selenium

The unit tests will be committed to source control and considered part of the deliverable, maintainable software product. Code integration process will include automatic builds and runs of unit tests if possible. If unit tests fail at the integration level, code promotion will be blocked, and the associated Jira issue status will be modified to reflect the need for further developer rework and retesting.

4.5 System Integration and Testing

The SwRI Team will test the R-ICMS at several levels to assure quality. The results of all the testing will be reviewed with FDOT prior to acceptance testing in the Test Readiness Reviews. The following summarizes the comprehensive development and testing approach that will be used for this project:

Unit Testing: Software developers will define and run unit tests during development. Once a module has successfully completed unit testing, the product will be promoted into the integration stream.

Integration Testing: Project staff will test modules in the integration stream using defined integration tests. Integration tests will be against derived requirements and will verify the performance of the software with other modules, such as the integration of a subsystem with its UI. Upon successful completion of integration testing, the software will be promoted into the Acceptance Test Procedure (ATP) stream. The results of R-ICMS Interface Test Cases will be provided to FDOT District 5.

Acceptance Testing: Project staff will perform “dry-runs” of the System Acceptance Test Procedures prior to submittal of the software for acceptance testing. The development team will run performance tests on the completed system. Failures of the ATP dry run will be noted, and the software returned to the developer for correction. The software is not ready for ATP until the ATP dry run completes without issues.

Performance Testing: Project staff will perform load/performance testing of the R-ICMS.

Configuration Audits. Prior to a release, the SwRI Team will perform configuration management audits to ensure the integrity of the software. The multi-step process consists of retrieving the source code from the Software Configuration Management (SCM) tool, building the software from the retrieved code, creating installers from the build, installing the software on a representative target system, and running acceptance tests against a newly installed system. This

process will assist in providing assurance that the release contains the functionality expected and meets requirements.

4.6 System Acceptance Test

This section describes the approach to be followed for participating in system qualification testing. System acceptance testing will be conducted in accordance with the PSEMP. The software engineer(s) will be responsible for supporting the testing activities. These activities will include assisting with the debugging process as problems are uncovered and correcting any software related problems. Subtopics to be addressed include:

- Independence in system acceptance testing
- Testing on the target computer system
- System acceptance test preparation
- Performing system acceptance test dry runs
- Performing system acceptance testing
- Revision and retesting
- Analyzing and recording system acceptance test results

SwRI will conduct System Acceptance Testing (SAT) in accordance with the FDOT reviewed and accepted Software Test Plans and Software Test Procedures. SAT will be conducted at the conclusion of all four iterations; each SAT will consist of three phases

- Test Readiness Review that will consist of a review of all preparations for SAT including test records from Unit Test, Integration Test and configuration audits.
- Acceptance Test that will consist of independent testing or oversight of running of the System Test Procedures in accordance with the System Test Plan.
- Correction Action Planning that will consist of a Hot Wash Up meeting that will review the results of the testing especially noted anomalies and plan any required corrective actions.

4.7 Preparing for Software Use

Following acceptance of the product of each software iteration, SwRI will restore the test system to a production state, verify the installation and turn the system over to FDOT for use. Each iteration will have a set of documents that describe the iteration. The Version Description Document and Software User Manual will be iteration specific and will be cumulative in content as they progress from numbering from 1 to 4.

- SwRI will re-install the system from the Version Description Document (VDD) or alternatively remove the SAT test data and connect the system with real data feeds for production use.
- SwRI will provide updated Software User Manuals (SUM) based on the functionality contained in the subject iteration.

4.8 Software Configuration Management

SCM will follow proven methods used by SwRI for FDOT in the SunGuide contract.

4.8.1 Configuration Identification

SwRI will implement a methodology for uniquely identifying baselined items under the R-ICMS contract. SwRI will assign to each baselined item a unique identifier incorporating iteration number and monotonically increasing numbers. The identifiers will be based on the suggested identification described in the contract Exhibit A Section 5 and Table 6. Section 5 states in part:

> Provide a unique numerical identification code for each deliverable for tracking purposes. It is suggested that a draft and final be indicated by a decimal value, for example, a preliminary draft work plan might initially be version 0.1, a subsequent draft might be identified as version 0.2 and the final document accepted by the DEPARTMENT would then be identified as version 1.0.

Software Releases

SwRI will identify each software release under this contract with unique configuration identifier.

- SwRI anticipates four major software releases under this contract, one for each iteration and each numbered according to its iteration number, i.e. 1 to 4.
- The VDD will identify specific document versions for documentation relevant to that version (e.g. SRS, Software User Manual, Software Test Plan, Software Test Procedure, Software Test Report, etc).
- The VDD will identify specific software versions for component modules that make up that version.
- Individual software modules will have their own identification numbering because they may have been reused from other software development efforts (e.g. open source projects, COTS, AAM or SunGuide).

4.8.2 Configuration Control

SwRI will implement Configuration Control processes to control and track changes. SwRI will establish a Jira based issue tracking system to be utilized for all issue tracking including:

- A **problem** or **bug** is any occurrence of deviation from expected outcomes, where the R-ICMS *is not performing* to defined specifications.
- A **change** is any occurrence of deviation from expected outcomes, where the R-ICMS *is performing* to specifications and the specifications are in error.

- An **enhancement** is any condition where a stakeholder (customer, user, developer …) finds an area that may be enhanced or improved; however, all *specifications are met* and they must be modified to incorporate the enhancement.

The SwRI team will take the following actions to control the configuration of the software in reaction to these three types of issues:

- Development managers will assign developers to correct problems or bugs during the contract without approval of FDOT. During the Operation, Maintenance and Support phase the team must have approval to install the revised software.
- The SwRI Team will complete a Change Order kept on line in the ASTLog spreadsheet for a modification, either a change or enhancement. The SwRI Team will provide a description of the proposed modification, proposed requirements and estimated cost. The team will make no effort to implement the change without FDOT approval. This falls under contract clause 3.6. Changes are not considered complete until documentation has been updated and the change accepted by FDOT.

4.8.3 Configuration Status Accounting

Configuration Status Accounting includes the process of recording and reporting configuration item descriptions (e.g., hardware, software, firmware, etc.) and all departures from the baseline during design and production. SwRI will implement Configuration Status Accounting in two ways.

1. SwRI will use the RMSIS to track approved requirements, deviations from the requirements, changes to the requirements, deleted requirements and when and by whom those approvals were given. The RTVM will then be used to track the approved requirements set to design, implementation, test and deployment.
2. SwRI will use the VDD associated with a software release to identify the configuration items including documents, code modules and other data that make up a release. The VDD will identify precise versions and dates of those configuration items.

4.8.4 Configuration Audits

SwRI will conduct Physical Configuration Audits and Functional Configuration Audits according to commonly used software configuration management practices.

Physical Configuration Audits. Prior to SAT, SwRI will develop a DRAFT VDD identifying configuration items that make up a release baseline. SwRI will audit the configuration against that baseline and generate / gather the related items from the identified baseline items. The collection of baseline items will be marked and changes to the baseline items prohibited prior to and during SAT. In this manner, FDOT will have confidence that what is tested are the appropriate baseline items.

Functional Configuration Audit. During SAT, the participants in the testing will only test those items identified as part of the baseline.

4.8.5 Packaging, Storage, Handling and Delivery.

At the conclusion of a successful Test Readiness Review, System Acceptance Test, Hot Wash Up, and delivered Corrective Action Plan(CAP), SwRI will collect those items identified by a Configuration Status Accounting Report (CSAR) into a deliverable product. The product will be delivered as a single release to FDOT by a means specified by FDOT, e.g. on releasable media (optical media, flash media, punch cards), zip file stored on the FDOT hosted SharePoint site, or other means.

4.9 Software Product Evaluation

Software Product Evaluations are described in the PSEMP Section 2.9 System Testing, Integration, Verification and Acceptance Planning and Section 3.14 System Acceptance.

4.10 Software Quality Assurance

The PSEMP Section 3.13 Quality Management details the Software Quality Assurance (SQA) processes that will be used during development of the R-ICMS. In summary of that, SQA processes will utilize the following processes:

- Formal customer reviews: including Software Requirements Walkthrough, PDD, and CDR
- Peer reviews: SwRI and EPIC will peer review all new code modules utilizing either Buddy Checks or Walk-Throughs
- Testing: all software components will be tested during development (Unit Testing), during integration (Integration Testing), system testing prior to Acceptance Testing (System Integration Testing) and during Acceptance Testing.
- Testing will include testing with the Modeling Engine to ensure that the two systems are properly integrated.

4.11 Corrective Action

SwRI will conduct a Hot Wash Up meeting following the Acceptance Test for each iteration. During the Hot Wash Up meeting, FDOT and SwRI will review the results of the acceptance test and the RTVM to identify requirements allocated to the iteration that were either not tested or failed testing. SwRI team will formulate a CAP that recommends a corrective action for each of those identified requirements. The corrective actions may be to remedy anomaly and retest, defer testing of the requirement to a later iteration or waive the requirement. CAP will contain a schedule for the corrective actions including recommended schedule for any recommended retesting.

4.12 Technical and Management Reviews

SwRI will conduct Management Reviews and Technical Reviews during the Software Development Lifecycle in accordance with the PSEMP Sections 3.11.2 Monthly Project Status Review and Section 2.5 Perform Technical Reviews.

4.13 Other Software Development Activities

This section describes the approach to be followed for other software development activities.

- Risk management is described in the PSEMP Section 2.6 Identify, Assess and Mitigate Risk and Section 3.6 Risk Management.
- Security and privacy is described in multiple approved requirements of the SRS
- Subcontractor management is described in the PSEMP Section 3.7 Subcontractor Management.
- Software Independent Verification and Validation (IV&V): no requirement for IV&V has been identified at this time. If FDOT identifies a requirement for IV&V and a role for SwRI as part of that IV&V effort, this document will be revised to describe how SwRI will satisfy that requirement.
- Coordination with associated developers: SwRI will coordinate its development activities with those of associated development activities as identified by FDOT. SwRI recognizes the importance of this coordination for successful integration of the R-ICMS and the ME efforts. Especially:
    - SwRI will expose the DFE APIs to the ME provider for ME provider access to the DFE data needed to support ME simulations. This access may either be through point to point Virtual Private Network access to the DFE APIs, through exposure of the APIs to the public internet by identified IP address, or through a sufficiently mature DFE installed in FDOT facilities.
    - SwRI will work to establish like access to the ME for integration of the DSS components (Expert Rules Engine, Evaluation Engine, SOT) to the ME for testing of the DSS.
    - SwRI recognizes the cooperative access must not expose either SwRI or the ME developer to unnecessary risks due to network intrusion by 3rd parties.
- Improvement of project processes. SwRI implements a software development process that has been assessed in compliance with Software Engineering Institute Capability Maturity Model Level 5 (high maturity). As such, the SwRI development team is committed to continual process improvement.

5 User Definitions

None.