Appendix H

Software Development Plan Template

This page is intentionally left blank.

Title Page
Document Control Panel
Table of Contents
List of Acronyms
Definitions

1. Overview

The Software Development Plan (SDP) establishes the software development approach, methodologies, tools, and procedures to be used during the analysis, design, development, testing, integration, deployment, and maintenance of the software for each of FDOT 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

This section contains a full identification of the system to which this document applies, including, as applicable, identification number(s), title(s), abbreviation(s), version number(s), and release number(s). The project SDP is managed and controlled in accordance with the FDOT's configuration management (CM) practices.

1.3 System Overview

This section provides a brief description of the operational concepts and products that will be developed using this SDP.

1.3.1 Operational Concept

1.3.2 Computer Software Configuration Items

1.4 Relationship to Other Plans

This section describes the relationship of the SDP to the other project management plans, including:

- Program management plans (PMPs)
- Systems engineering management plans (SEMPs)
- Hardware development plans (HDPs)
- Configuration and data management plans (CDMPs)
- Quality management plans (QMPs)
- Quantitative management plans

2. Reference Documents

This section lists the number, title, revision, and date of all documents referenced in this plan.

- FDOT documents
- Project plans
- Operational procedures
- Engineering instructions

3. Overview of Software Development Planning

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

- Requirements and constraints on the system and software
- Requirements and constraints on project documentation
- Position of the project in the system life cycle
- Selected program/acquisition strategy or any requirement constraints on it
- Requirements and constraints on project schedules and resources
- Training required for project engineers, both initial and ongoing
- Other requirements and constraints, such as project security, privacy, methods, and standards, etc.

3.1 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 configuration items (CSCIs). Any additional system constraints are documented in the project SEMP.

3.2 Project Documentation

This section defines the tool(s) used to produce the project documentation (i.e., Microsoft Word, Microsoft Excel, etc.) and how document configuration will be controlled.

3.3 System Life Cycle

This sections describes the life-cycle model used across FDOT ITS projects. The model encompasses the entire development process from requirements analysis through deployment.

3.4 Schedules and Resources

This section provides the detailed schedule for the project. The schedule depicts the assigned personnel as well as the critical path and any critical dependencies.

3.5 Training Requirements

This section defines the software development training requirements for the project software engineers.

4. General Software Development Activities

4.1 Development Process

This section describes the software development process to be used. It also identifies planned software builds, their objectives, and the software development activities to be performed in each build.

4.2 Development Methods

This section identifies the software development methods (e.g., object-oriented methodology, etc.) to be used. It includes descriptions of the manual, automated tools, and procedures to be used in support of these methods.

4.3 Product Standards

This section provides reference to the standards to be followed for representing requirements, design, code, test cases, test procedures, and test results.

4.4 Reusable Products

4.4.1 Incorporating Reusable Products

This section describes the approach to be followed for identifying, evaluating, and incorporating reusable software products, including the scope of the search for such products and the criteria to be used for their evaluation.

4.4.2 Developing Reusable Products

This section describes the approach to be followed for identifying, evaluating, and reporting opportunities for developing reusable software products.

4.5 Critical Requirements

This section describes the approach to be followed for handling requirements designated critical. For example:

- Safety assurance
- Security assurance
- Privacy assurance
- Assurance of other critical requirements

4.6 Computer Hardware Resource Utilization

This section describes the approach to be followed for allocating computer hardware resources and monitoring their utilization.

5. Detailed Software Development Activities

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

5.1 Project Planning and Oversight

This section describes the approach to be followed for project planning and oversight.

- Software development planning
- Computer software configuration item test planning
- System test planning
- Software installation planning
- Software transition planning
- Following and updating plans, including the intervals for management review

5.2 Establishing a Software Development Environment

This section describes the approach to be followed for establishing, controlling, and maintaining a software development environment.

- Software engineering environment
- Software test environment
- Software development library
- Software development files

5.3 System Requirements Analysis

This section describes the approach to be followed in system requirement analyses. System requirements analysis processes will be performed as described in the project SEMP.

- Analysis of user input
- Operational concept
- System requirements

5.4 System Design

This section describes the approach to be followed for participating in system design.

- Systemwide design decisions
- System architectural design

5.5 Software Requirements Analysis

This section describes the approach to be followed for software requirements analysis. It also defines the entrance and exit criteria for the software development process.

5.6 Software Design

This section describes the approach to be followed for software design. It also defines the entrance and exit criteria for the software design process.

- Software architecture/preliminary design
- Software detailed design

5.7 Software Implementation, and Test / Code and Unit Testing

This section describes the approach to be followed for software code and unit testing. It also defines the entrance and exit criteria for the software code and unit test processes.

- Software code
- Unit testing preparations
- Performing unit testing
- Revision and retesting
- Analyzing and recording unit test results

5.8 Software Integration and Testing

This section describes the software integration activities that focus on integrating all the software components into a software element that will be further integrated with the other elements to form the systems.

- Preparing for integration and testing
- Performing integration and testing
- Revision and retesting
- Analyzing and recording integration and test results

5.9 System Integration and Testing

This section describes the approach to be followed for participating in system integration and testing (SIT). System integration testing will be conducted in accordance with the project SEMP. 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 SIT
- Testing on the target computer system
- Preparing for SIT
- Performing SIT dry runs
- Performing SIT
- Revision and retesting
- Analyzing and recording SIT results

5.10 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 project SEMP. 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

5.11 Preparing for Software Use

This section describes the approach to be followed for preparing for software use.

- Preparing the executable software
- Preparing version descriptions for user sites
- Preparing user manuals
- Installing at user sites

5.11.1 Preparing for Software Transition

This section describes the approach to be followed for preparing for software transition. Software transition consists of the preparation of all software files, including source and executables, for delivery, and updating the final version descriptions. All files will be identified and verified via recompilation and/or reference checking after that component has successfully passed the required testing and been placed into controlled libraries for inclusion in the software build.

- Preparing the executable software
- Preparing source files
- Preparing version descriptions for the support site
- Preparing the "as-built" CSCI design and other software support information
- Updating the system design description
- Preparing support manuals
- Transitioning to the designated support site

5.12 Software Configuration Management

This section describes the approach to be followed for software configuration management (CM). The CM processes and procedures are described in the CDMP.

- Configuration identification
- Configuration control
- Configuration status accounting (CSA)
- Configuration audits
- Packaging, storage, handling, and delivery

5.13 Software Product Evaluation

This section describes the approach to be followed for software product evaluation. Software product evaluation will be accomplished according to the program schedules established for the project.

- In-process and final software product evaluations
- Software product evaluation records, including items to be recorded
- Independence in software product evaluation

5.14 Software Quality Assurance

This section describes the approach to be followed for software quality assurance (QA).

- Software QA process and product evaluations
- Software QA records, including items to be recorded
- Independence in software QA

5.15 Corrective Action

This section describes the approach to be followed for corrective actions (i.e., problem/change reports, etc.).

5.16 Technical and Management Reviews

This section describes the approach to be followed for technical and management reviews.

5.17 Other Software Development Activities

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

- Risk management, including known risks and corresponding strategies
- Software management indicators, including indicators to be used (i.e., metrics)
- Security and privacy
- Subcontractor management
- Interface with software independent verification and validation (IV&V) agents
- Coordination with associate developers
- Improvement of project processes

6. Schedules and Activity Network

This section presents the schedule information or refers to detailed software schedules published for the project, including:

- Schedule(s) identifying the activities in each build and showing initiation of each activity

- Availability of draft and final deliverables, and other milestones; and completion of each activity
- An activity network, depicting sequential relationships and dependencies among activities and identifying those activities that impose the greatest time restrictions on the project

7. Program Organization and Resources

This section describes the project organization and resources to be applied in each software build, increment, iteration, or prototype.

8. Acronyms and Notes

This section contains any general information that aids in understanding this document. This section will contain an alphabetical listing of all acronyms and abbreviations, and their meanings as used in this document, and a list of any terms and definitions needed to understand this document.

This page is intentionally left blank.