



## System Development and Life-Cycle Management (SDLCM) Methodology

<b>Subject</b> Software Development Plan	<b>Type</b>	Standard
	<b>Identifier</b>	S-1057
	<b>Effective Date</b>	May 2002
	<b>Revision No.</b>	1

Approval

CISSCO Program Director

### A. PURPOSE

This standard specifies the content and format requirements for a Software Development Plan (SDP).

This document is a Plan for the software development/enhancement effort of a Project. As such, it should identify distinct software design activities; specify how and when they are to be done; and describe the necessary development environment, include ancillary processes. This Plan should provide the Technical Manager the information needed to accomplish an assigned task and provide the Overall Project Manager with information necessary to manage the overall Project and reflect the software effort appropriately in the Project Management Plan.

Sections in this document may refer to other SDLCMM documents or an appendix for required information.

### B. APPLICABILITY

The *Software Development Plan* (SDP) is a management document that describes the activities necessary for the management, design, testing, quality assurance and configuration management of software. Because the SDP is a Plan, it identifies the activities and outlines how they are to be performed. It is typically drafted during Component 1 and finalized late in the system design phase (Component 3), that is, after a system design has been decided upon. Its intended audience is all Project personnel.

The SDP is a Plan for developing the information technology portions of the overall system defined by the System Requirements Specification(SRS). The SDP expands upon the information provided in the PMP. It also describes the methods for software requirements definition and software design.

### C. REFERENCE PUBLICATIONS

The following publications contain related information:

- *SDLCM Methodology Handbook*, Component 1, Outputs and Deliverables
- *SDLCM Methodology Handbook*, Component 2, Outputs and Deliverables

<b>Subject</b> Software Development Plan	<b>Type</b>	Standard
	<b>Identifier</b>	S-1057
	<b>Effective Date</b>	May 2002
	<b>Revision No.</b>	1

- *SDLCM Methodology Handbook*, Component 3, Outputs and Deliverables
- *SDLCM Methodology Standard*, S-1051, Project Charter
- *SDLCM Methodology Standard*, S-3051, System Requirements Specification
- *SDLCM Methodology Standard*, S-3052, Current System Assessment Document
- *SDLCM Methodology Standard*, S-3053, System Operations Concept
- *SDLCM Methodology Standard*, S-1052, Project Management Plan
- *SDLCM Methodology Standard*, S-3091, Software Engineering Notebook
- *SDLCM Methodology Standard*, S-5151, Test Plan
- *SDLCM Methodology Standard*, S-3162, Context Diagrams
- *SDLCM Methodology Standard*, S-3151, Data Models
- *SDLCM Methodology Standard*, S-1055, Development and Maintenance Environment Products Installation Plan
- *SDLCM Methodology Standard*, S-5051, Tactical Integration Plan

#### **D. STANDARD**

The Software Development Plan, provides the detailed activities and schedules for designing, coding, integrating, and testing new, legacy and COTS software modules to provide a full functionality of the software for the Project. It is developed on Projects that include software development/enhancement or integration after Project requirements, as identified in Component 1 of the SDLCM Methodology, are assigned to be addressed through automated solutions. The SDP is updated and finalized as requirements and design solutions are refined in Component 3 of the SDLCM Methodology.

The level of detail and content of the SDP may vary with the Project, development approach, or management style. Tailor this standard to be consistent with the size, scope, and complexity of the Project. Add sections and subsections for special topics. Sections and subsections that are not applicable should *not* be deleted; they should indicate "Not Applicable." Include a level of detail that allows for successful management of the Project.

Some of the information contained in the SDP is based on information developed for the Project Charter, Alternatives Analysis, Current System Assessment Document, System Operations Concept and the System Requirements Specification. Copy and build on this information as appropriate; do not redevelop it. When appropriate, use references and pointers to other documents and Plans rather than repeating material unnecessarily. However, repeat important material as necessary to clarify or to emphasize aspects of the Plan.

The following paragraphs describe the content of each section of the SDP.

<b>Subject</b> Software Development Plan	<b>Type</b>	Standard
	<b>Identifier</b>	S-1057
	<b>Effective Date</b>	May 2002
	<b>Revision No.</b>	1

## 1. INTRODUCTION

This section contains the Plan for development of the software component of a Project. Identify the CPIC Project identification number, where appropriate, and uniquely assigned business application system(s) number, title(s) and acronym(s), when applicable, for the system/application to which the Software Development Plan applies, define the scope of the SDP, specify the applicable references to other Project documents (such as the Project Charter, Project Management Plan, Current System Assessment Document, Alternatives Analysis, System Operations Concept, System Requirements Specification, and define any terms unique to this Plan or to the Project.

### 1.1 Background

Describe the background for the Project from the client's perspective. Explain why the development of software is being initiated, its importance to the client, the specific results or long-term objectives that are desired, and contribution that the software will contribute to the overall strategy. Include the business problem that is to be solved or the software to be developed in sufficient detail to support planning.

### 1.2 Objectives

Specify the Project objectives that this software is to support from the customer's perspective, the critical functions the software should achieve, and the quantifiable criteria the software must meet to succeed. Identify high-level requirements or, if applicable, reference an existing document (such as the System Requirements Specification) that contains the requirements.

### 1.3 Scope

Define all aspects of the software development addressed by this Plan.

Describe any external influences and impacts, such as all organizations involved in implementing the Software Development Plan (infrastructure support for automation efforts, etc.), data and system interfaces, customer or other business needs, and regulatory requirements that are addressed by the software.

Refer to the software scope contained in the Project Charter, Project Management Plan, Current System Assessment Document, System Operations Concept and the System Requirements Specification, as appropriate. If necessary, copy important information from these documents and build on it to clarify or emphasize aspects applicable to this Plan.

Specify any constraints, or restrictions, associated with the software development effort. A constraint may relate to approach, priorities, personnel, time, technologies, environments, tools, and techniques, or other aspects of the software development effort. In addition, discuss how management will control identified constraints to ensure success.

<b>Subject</b> Software Development Plan	<b>Type</b>	Standard
	<b>Identifier</b>	S-1057
	<b>Effective Date</b>	May 2002
	<b>Revision No.</b>	1

## 1.4 Assumptions

Specify the assumptions used to establish the software development estimates, Plans, and approach. Include each identified open issue if the software development effort is to continue while issues remain unresolved. Include the degree of criticality in the description of each assumption.

Refer to the assumptions contained in the Project Charter, Project Management Plan, Current System Assessment Document, System Operations Concept and the System Requirements Specification, as appropriate. If necessary, copy important information from these documents and build on it to clarify or emphasize aspects applicable to this Plan.

## 1.5 Applicable Documents

Specify any documentation used to support the creation of the SDP, to provide additional information regarding the software aspects of the Project, or to be used in completing these aspects of the Project, including applicable standards and process documentation.

List the documents. Cite documents by publisher or source; document number or other unique identifier (if any); title, version or release designator (if any); and date. Note, that the particulars for any document(s) actually cited in the text should be listed in the references section at the end of the document.

## 1.6 Document Overview

Summarizes the purpose and contents of the SDP. Include documentation of tailoring decisions. Discuss the organization of the SDP. Describe each major section of the Plan in terms of its basic content and relationship to the Project.

Describe how the Plan will be maintained throughout the life of the Project (for example, via document change notices). Identify Project milestones at which the SDP will be updated.

## 1.7 Relationships to Other Plans

Describes the relationship, if any, of the SDP to related Project management Plans.

## 1.8 Definition of Terms

List the terms/acronyms and corresponding definitions specific to the software development/enhancement Planning process and the software aspects of the Project.

<b>Subject</b> Software Development Plan	<b>Type</b>	Standard
	<b>Identifier</b>	S-1057
	<b>Effective Date</b>	May 2002
	<b>Revision No.</b>	1

## 2. APPROACH

Describe the framework for managing and integrating the software and specify the overall schedule and organization critical to the success of the development effort. Cost issues are the responsibility of the Overall Project Manager to coordinate with the Technical Project Manager.

Describe the technical approach to be followed for this software development effort. Identify the life-cycle model (for example, the waterfall, incremental, evolutionary, package-based life-cycle models) that will be used (Life-cycle models are addressed in the *SDLCM Methodology Handbook*). Include a high-level description of the test approach and the levels of testing that will be performed to ensure that the software meets its requirements.

## 3. SOFTWARE DEVELOPMENT MANAGEMENT

This section defines and describes the Project's approach to managing software development and integration, including the detailed activities and schedules for designing, coding, integrating, and testing new, legacy, and COTS software modules to provide the full functionality of the software for the Project. It is included for Projects that involve software development/enhancement or integration. It is drafted after Project/System requirements have been identified in Component 1 and assignment of a solution has been made for the automation component of the effort. It is further updated and finalized as requirements are refined and design is defined in Component 3 of the SDLCM Methodology.

Consistent with the software life-cycle model identified in Section 2, identify the software development methods, techniques, procedures, and standards for developing, enhancing, or integrating COTS products for the Project. Identify any tools that will be used.

Briefly describe how the software products produced by the Project will fit into the overall system.

### 3.1 Software Development Organization and Responsibilities

Describes the software development organization, for example, the authority and responsibilities of each organization, the personnel necessary to complete the software development/enhancement efforts and how this fits into the Overall Project organization.

#### 3.1.1 DEVELOPMENT TEAM PERSONNEL

Explain and illustrate using diagrams how the development team will organize activities and personnel to carry out the software development activities of the Project, including types and numbers of personnel assigned, reporting relationships, and team members'

<b>Subject</b> Software Development Plan	<b>Type</b>	Standard
	<b>Identifier</b>	S-1057
	<b>Effective Date</b>	May 2002
	<b>Revision No.</b>	1

authorities and responsibilities. Where applicable use roles as defined in *SDLCMM Handbook, Appendix A*.

Identify personnel resources required to perform software-related activities for the Project. Include the following information (use a table, if applicable):

- Activities to be performed
- Roles needed
- Any special skills needed
- The personnel assigned to those roles

### 3.1.2 INTERFACING GROUPS

Identify interfacing groups or contractors, points of contact, and the responsibilities of each group/contractor. This should include interfaces between the development effort and the Executive Sponsor, Overall Project Manager, Technical Project Manager, Business Advocate, and any other entity that has concern or interest or involvement in the outcome of the Project. It is imperative that the Plans for, and status of, the development effort be communicated in a structured manner to the entire Project. Of particular interest are the interfaces in the following areas

- Between the development effort and the Business Advocate to ensure that NRC testing is adequately addressed
- Status reporting from the development effort to/through the Technical Project Manager to the Overall Project Manager and Executive Sponsor
- Intra-NRC interfaces

Use a table or diagram to illustrate these interfaces.

## 3.2 Software Development Technical Approach

Divided into separately numbered sections that describe the methodology, standards and procedures to be followed when designing software for this Project.

### 3.2.1 ACTIVITIES, TOOLS AND PRODUCTS

For the software effort's (automated portion of the Project) selected life-cycle model, identify the major activities to be performed, the software development methods, techniques, procedures, standards and tools that will be applied, and the products that will be developed during each life-cycle phase.

#### 3.2.1.1 Design Methods

Defines the design methods to be used for the software on this Project. The definition should include the following:

- General description of the methods

<b>Subject</b> Software Development Plan	<b>Type</b>	Standard
	<b>Identifier</b>	S-1057
	<b>Effective Date</b>	May 2002
	<b>Revision No.</b>	1

- Detailed specification of the methods (by reference, if possible)
- Additional methodology requirements, for example, reusability features, generic features
- Examples using this methodology, if applicable

This section may appear as an appendix to the SDP or may reference a separate methodologies document.

#### 3.2.1.2 Software Engineering Notebook

Defines the Plans — including the organization responsible — for the creation and maintenance of the Software Engineering Notebook(s) (SENs). The format and contents of the SENs are defined, as well as the procedures for maintaining the SENs. Special procedures for use in this Project should be referenced in this paragraph.

#### 3.2.1.3 Software Programming Standards

Describes or references the standards for design language, program structure, coding and testing.

#### 3.2.2 Non-Developmental Software

Describes each non-developmental software item, such as commercially available or reusable software that is incorporated into the deliverable software. Also describes the rationale for the use of each non-developmental software item.

#### 3.2.3 Integration and Testing

Describe the process for integrating the builds and releases. Describe the Plan for all informal and formal testing activities.

Define the scope and hierarchy for testing. Describe the Plan for all formal qualification testing activities. State the level of independence of the various test teams. Summarize any critical high-level acceptance testing criteria.

##### 3.2.3.1 Resources and Organization

Identifies the organizations responsible for the following areas:

- Formal qualification testing
- Determining the personnel necessary to perform testing
- Identifying the authority and responsibilities of each organization

This section also describes the resources necessary for the above areas.

<b>Subject</b> Software Development Plan	<b>Type</b>	Standard
	<b>Identifier</b>	S-1057
	<b>Effective Date</b>	May 2002
	<b>Revision No.</b>	1

Levels of Testing - Describes the Planned levels of testing, e.g., Software Integration and Testing, System Integration and Testing and Acceptance Testing.

Software Implementation - Describe the implementation strategy to be used on the Project. Describe the builds and releases to be implemented and their functionality. Describe the integration of COTS products with developed products. Identify the tools, workstations, and local area networks that will be used by the development team. Define the problem reporting and corrective action process that will be employed.

Risk Factors - State software development specific risks, if applicable, and indicate how they have been, or will, be mitigated. Risk management may be handled in this document or referenced in a companion Risk Management Plan.

Classes of Testing - Identifies the classes of testing to be performed, for example, regression tests, stress tests, hardness tests, performance tests, load tests.

### 3.2.3.2 Software Test Environment

Describes the software test environment for all formal testing.

Existing Test Tools - Describe known test tools that are currently on hand, or which have been applied previously, which will be used for Planned testing. List special or peculiar requirements that must be addressed or satisfied.

New Test Tools - Describe known test tools that are Planned for acquisition, or which have not been applied previously, which will be used for Planned testing. List special or peculiar requirements that must be addressed or satisfied.

Configuration Requirements - Describe test bed configuration requirements which will be necessary for all levels Planned testing, e.g., development, integration, system and acceptance.

Personnel - Describe personnel requirements which will be necessary for all levels of Planned testing.

Test Hardware - Describe the hardware, including special and/or peculiar requirements, that is necessary to accomplish Planned testing.

Test Environment Schedule Requirements - Describe the interplay of time availability for test environment assets. Discuss known and/or perceived constraints and risks, including risk mitigation, based on non-availability of one or more test environments.



<b>Subject</b> Software Development Plan	<b>Type</b>	Standard
	<b>Identifier</b>	S-1057
	<b>Effective Date</b>	May 2002
	<b>Revision No.</b>	1

### 3.2.3.3 Test Planning Assumptions and Constraints

Describes any assumptions that were made in test Planning and any constraints imposed upon formal qualification testing.

## 3.3 Software Development Management Approach

Describe items that affect the management approach, include software effort priorities within the overall Project priorities.

### 3.3.1 SOFTWARE DEVELOPMENT RESOURCE REQUIREMENTS

Specify estimated levels of resources required, including:

- Estimates of system size using appropriate units of measure (for example, new and reused lines of code and modules, number of user-interface screens, number of database transactions)
- Staff effort (managerial, programmer, and support, including contractors and NRC staff), by software component (subsystem, object, or unit). Include all necessary review times for deliverables, structured walkthroughs etc
- Training requirements
- Computer resources
- Software risk management - describe the procedures for managing areas of risk to ensure successful delivery of software to ensure Project completion:
  - Identification of the areas of risk to successful software delivery
  - Identification of the constituent risk factors that contribute to the potential occurrence of each risk
  - Documentation of procedures for monitoring the risk factors and for reducing the potential occurrence of each risk
  - Identification of contingency procedures for each area of risk (as appropriate)
- Security - Describes the Plans for implementing security requirements.
- Reviews - Describes the internal procedures for preparing and conducting reviews.
- Project-Level CM Library - Described the Project-Level CM Library, which is used to store and maintain versions and control changes to software, hardware and associated documentation and Project-level application system documentation. Includes a description of the procedures and methods for establishing and implementing the library and/or environment and the access and control procedures for data stored in the Project-Level CM Library. The Project-Level CM Library is defined in the Configuration Management Plan.

<b>Subject</b> Software Development Plan	<b>Type</b>	Standard
	<b>Identifier</b>	S-1057
	<b>Effective Date</b>	May 2002
	<b>Revision No.</b>	1

- Corrective Action Process - Describes the corrective action process to be implemented.

### 3.3.2 SOFTWARE DEVELOPMENT MILESTONES AND SCHEDULES

Specify the work to be done, who will perform it, its interdependencies, and when it will be completed, including the following:

- Development life cycle (the start and finish dates)
- Build and release dates
- Delivery dates of required external interfaces
- Schedule for integration of externally developed software and hardware
- List of data, information, documents, software, hardware, and support to be supplied by external sources and delivery dates
- List of data, information, documents, software, and support to be delivered to NRC and delivery dates
- Schedules for reviews (internal and external)
- Facilities - Provides a description of the facilities to be used for the effort. Highlights the locations of Project-specific resources, such as the software engineering environment and the software test environment.

Use tables, if appropriate. Include a discussion of the estimation methods or rationale used.

### 3.3.3 SOFTWARE DEVELOPMENT MEASURES

Identify the indicators that management will use to monitor progress and product quality. Measures, such as software size (i.e., lines of code, number of objects, function points, number of requirements, number of screens, etc.), requirement satisfaction, requirements/design stability, effort in staff months, duration in months, pages of documentation produced, annual maintenance cost, average staff size, action item status/problem reporting, test progress, assist the manager in determining whether Projected schedules, cost, and quality will be met and also show whether changes are required. Indicators may include:

- Standard earned-value systems aid in analyzing the rate of resources consumed compared to Planned completed products.
- Tracking changes made to controlled source libraries aids in identifying unstable requirements or design.
- Tracking reported versus fixed discrepancies found in testing aids in gaining insight into software reliability. Reporting is addressed at both the software development/enhancement effort level and the overall Project levels.

<b>Subject</b> Software Development Plan	<b>Type</b>	Standard
	<b>Identifier</b>	S-1057
	<b>Effective Date</b>	May 2002
	<b>Revision No.</b>	1

## REFERENCES

List all cited references.