System Development and Life-Cycle Management (SDLCM) Methodology

Subject	Type	Standard
Logical Design Document	Identifier	S–3171
	Effective Date	June 2002
	Revision No.	3

Approval _____

CISSCO Program Director

A. PURPOSE

This standard specifies the content and format requirements for a Logical Design Document.

B. APPLICABILITY

A Logical Design Document is required for all NRC projects, subject to the SDLCM Methodology, that include software development or integration.

The Technical Project Manager and members of the Development Team responsible for system or software architecture use this standard for documenting the design of the logical solution; the Overall Project Manager uses this standard to identify and isolate those procedure, policy, and guidance materials that must be developed by the non-software Project staff; key managers and quality assurance personnel use it for reviewing the Logical design; and the Executive Sponsor uses it when approving the Logical Design Document. The Logical Design Document is made available to all members of the project team, preferably in electronic form. This standard is not required for maintenance activities; however, the update of the PDD is required for perfective maintenance activities.

C. REFERENCE PUBLICATIONS

The following publications contain related information:

- *SDLCM Methodology Handbook*, Component 3, Design the Solution
- *Systems Development CASE Tool Guidelines*, Systems Development and Integration Branch (SDIB) Office of Information Resources Management (OIRM), September 12, 1995
- *Standards and Conventions*, SDIB OIRM, August 28, 1995

- *SDLCM Methodology Standard*, S–1052, Project Management Plan
- *SDLCM Methodology Standard*, S–3054, Alternatives Analysis Document
- *SDLCM Methodology Standard*, S–3151, Data Models
- *SDLCM Methodology Standard*, S–3161, Process Models
- *SDLCM Methodology Standard*, S–3162, Context Diagrams
- *SDLCM Methodology Standard*, S–3163, Data Flow Diagrams
- *SDLCM Methodology Standard*, S–xxxx, Entity Relationship Diagrams
  SDLCM Methodology Standard, S–3351, Data Dictionary
- *SDLCM Methodology Standard*, S–3172, Physical Design Document

D. STANDARD

The system architecture is selected during the logical design process. The Logical Design Document presents that architecture at a level of detail sufficient to begin detailed physical design activities. It translates the system's requirements, contained in the System Requirements Specification (SRS), into the functions to be performed by the hardware, software, and firmware components of the system and identifies necessary supporting policy, guidance and procedures documents. It shows how the various components will work together to meet the operational requirements.

Tailor this standard as needed to be consistent with the size, scope, and complexity of the system. Add sections and subsections for special topics. Sections and subsections that are not applicable should *not* be deleted; they should indicate "not applicable."

Some of the information contained in the Logical Design Document is based on information developed for the SRS. 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 design.

The following paragraphs describe the content of each section of the Logical Design Document.

1. INTRODUCTION

1.1 Background

Identify the system being designed. Using the SRS as a starting point and with references to it used to minimize repetition, describe the approach selected and how this design will solve the problem presented in the SRS. Describe how this design was selected. Briefly identify any alternatives that were considered but rejected. Identify any external constraints that were imposed (e.g., SYBASE must be the DBMS used as it is the NRC standard).

1.2 Objectives

Specify the objectives of the logical design, the critical functions it should achieve, and the quantifiable criteria it must meet to succeed. Refer to the project's business objectives contained in the SRS and expand on them as needed.

1.3 Scope

Identify and characterize the system, its software and architectural complexity, where it will be deployed, and the number of expected users and user sites.

State whether the project is a new, enhanced, or integrated or migrated version of an existing system.

Specify the high-level requirements to be satisfied by the system. Refer to the SRS for additional requirements to be satisfied by the system.

1.4 Assumptions

Discuss any assumptions that have been made about the logical design and discuss the risks, if any, associated with those assumptions.

1.5 Applicable Documents

Specify any documentation used to support creation of the Logical Design Document, to provide additional information related to the project or system, or to be used in designing or developing the system, 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: Any documents actually cited in the text should also be listed in the References section at the end of the document.

1.6 Overview

Discuss the organization of the Logical Design Document. Describe each major section of the document in terms of its basic content.

Describe how the Logical Design Document will be maintained throughout the life of the project (for example, via document change notices).

2. APPROACH

2.1 Design Methods

Provide a high-level summary of the method(s) used to create and evaluate the proposed logical design for the system. Describe any constraints on the design.

Discuss how the design method accounts for the use or integration of commercial off-the-shelf (COTS) products. Specifically, identify and discuss any directives or strategies for using COTS products. Specify COTS packages to be documented, analyzed, enhanced, or modified.

2.2 Design Alternatives

If applicable, provide a brief description of alternative design architectures and decomposition considered, including the one selected. Discuss the results of the analyses of candidate designs and the criteria used for selection.

2.3 Design Studies

If applicable, provide information on analysis, modeling, and trade studies performed during the logical design effort. Show how the results of the studies (such as workload; performance; reliability, maintainability, and availability; and database management system) demonstrate that the logical design will meet requirements. List any assumptions used and present the results.

2.4 Design Issues

Provide an overview of unresolved issues in the logical design. Discuss any to-be-determined (TBD) issues in the requirements and interfaces, their status, and what steps are planned or need to be taken to resolve them.

Discuss any risks or uncertainties in the hardware and software design. Describe the studies and prototyping efforts planned to resolve them during physical design.

3. LOGICAL MODEL

Provide an overview of the design of the software and hardware elements of the solution. Discuss the external interfaces to the system. Refer to any agreements concerning external interfaces. Provide a brief description of the high-level processes or subsystems that make up the logical design of the solution. Note: These are the processes that have been identified in the Level 1 data flow diagram.

3.1 System Architecture

Provide a high-level description of the architecture, an end-to-end data flow diagram, and a diagram(s) that identifies the internal and external interfaces for the proposed logical design. Include the context diagram produced for the SRS and update it if necessary. Include any applicable interfaces to corporate data, legacy systems, and users in other organizations or locations.

If applicable, describe the characteristics of the computer network, network components, and operating systems to be used.

3.2 Logical Data Model

Discuss the project's data modeling activities, including the identification of corporate data that must be accessed or manipulated by the system, any data that must be added to the corporate databases, capacity and archiving requirements for the data, and data security requirements.

3.2.1 ENTITY RELATIONSHIP DIAGRAM

Include a copy of the project's entity relationship diagram (ERD) and the entity descriptions contained in the data dictionary. If the project is using a computer aided software engineering (CASE) tool, refer to the on-line repository or encyclopedia where the logical data model and dictionary are stored and updated as needed.

If the CASE tool used by the project uses any conventions different from those described in Section 3.2, Data Model, of the *Systems Development CASE Tool Guidelines*, or Section 3, Logical Model Object Names, of the *Standards and Conventions* document, identify the conventions used and the project's approval to use the tool.

3.2.2 DATA MAPPING AND NAVIGATION TO LEGACY SYSTEMS AND SCREENS

Use a table to document the mapping between system data entities and corporate or legacy system data entities. Identify how the system will act on (read, manipulate, or update) these data entities.

Specify how the system will navigate to legacy systems to obtain needed data. If special screens must be developed so that the proposed system may access the legacy data, provide a first-cut design of the screens.

Identify data entry and display requirements and provide the initial design for these items. Discuss screen and dialog performance requirements that will be satisfied by the design.

> Note: The navigation process and any new screens or displays will be designed in greater detail as a part of physical design.

3.3 Logical Process Model

Discuss the project's process modeling activities, including the business procedures, practices, or decision-making processes that are to be automated and any transaction performance requirements, such as volume, speed, and frequency of transactions. Identify any processes that will not be automated and ensure that these are documented as such and that they are appropriately addressed in Section 5.1 and are reflected in updates to the PMP.

Copies of the project's data flow diagrams, also known as logical process models, and the process descriptions are included in the subsystem-by-subsystem description

contained in Section 4 of this document, so this section should refer the reader to Section 4 and to the on-line repository or encyclopedia where this information is stored and updated.

If the CASE tool used by the project uses any conventions different from those described in Section 3.3, Process Model, of the *Systems Development CASE Tool Guidelines*, or Section 3, Logical Model Object Names, of the *Standards and Conventions* document, identify the conventions used and the project's approval to use the tool.

4. SYSTEM DESCRIPTION

Include a copy of the Level 1 DFD of the logical process model. For each process described in the Level 1 DFD, identify subsystems and configuration items (CIs), both those that will be implemented in software (including COTS products) and those that will be implemented in hardware or firmware. Provide high-level architectural hierarchy or other block diagrams that show the relationships among the subsystems. Provide a brief, high-level narrative description of these diagrams that identifies the hardware and software subsystems and CIs, the functions they perform, and how they interact. For a software subsystem based on a COTS product, discuss the subsystem in terms of the COTS product as a whole, rather than in terms of the processes provided within the COTS product. The narrative information may be provided in bulleted lists or a table as appropriate.

For each process, provide a subsystem description in the format shown below.

4.1 Subsystem Descriptions

4.1.1 NAME OF SUBSYSTEM 1

Provide a functional description of the subsystem and a high-level review of the requirements met by the subsystem. Use the subsystem's process description. Include a copy of the DFD(s) that decompose this subsystem to show the software CIs contained in the subsystem. If applicable, provide a block diagram showing the hardware CIs contained in the subsystem. Discuss the interfaces between this subsystem and each of the other subsystems in the system. Discuss the interfaces among the CIs in the subsystem.

4.1.1.1 SOFTWARE Configuration Items

Provide a summary of the software CIs.

4.1.1.1.1 Software CI 1 Description

Provide a functional description of the software CI and a high-level review of the requirements met. Use the CI process description. Identify the source of the CI and whether it is to be developed, purchased, or acquired from a combination of sources. List files and databases accessed by the CI.

4.1.1.1.*n* Software CI *n* Description

Provide the same information for each additional software CI in subsystem 1 as for software CI 1.

4.1.1.2 HARDWARE Configuration Items

Provide a summary of the hardware CIs.

4.1.1.2.1 Hardware CI 1 Description

Provide a functional description of the hardware CI and a high-level review of the requirements met. Identify the source of the CI and whether it is to be developed, purchased, or acquired from a combination of sources.

4.1.1.2.*n* Hardware CI *n* Description

Provide the same information for each additional hardware CI in the subsystem as for hardware CI 1.

4.1.*N* SUBSYSTEM *N* DESCRIPTION

Provide the same information for each additional subsystem as for subsystem 1.

4.2 External Interfaces

Provide a description of each external interface (for example, interfaces to the network topology, another system, subsystem, or CI). Identify the name of the interface, its type (for example, hardware interface, data file), purpose, and frequency of interchange. Describe the structure and organization of the data. Reference the system's data model to provide the format of the data components, including such items as data element type, description, representation, range of values, and units. Estimate the amount of data crossing the interface and estimate the imposed data storage requirements.

If applicable, refer to or include an external interface diagram for each interface.

4.3 Internal Interfaces

Provide a description of each interface between and among the components of a CI. Identify the name of the interface and its type, purpose, and frequency of interchange. Describe the organization of the data (for example, record and block structure). Define the preliminary format of the data components, including data element type, description, representation, range of values, units, etc.

5. SYSTEM OPERATIONS DESIGN

Provide an operational overview of the system.

5.1 Operations Scenarios

Discuss the design in terms of the operational scenarios for each major product produced. Show how the elements of the design work together to facilitate the production of the product. Discuss how the design meets operational requirements such as performance and data quality. Review and identify changes or needs for creation of policy, procedures, and/or guidance necessary to define and support business area processes whether automated or non-automated to be consistent with these operational scenarios.

5.2 User-System Interface

Provide a description of the look and feel of the user or system interface. Discuss how the users will interact with the system to perform various functions.

APPENDIX A. REQUIREMENTS-TO-DESIGN TRACEABILITY

Include a matrix that traces each subsystem and CI to its source in the System Requirements Specification (SRS).

APPENDIXES (If needed)

If design analysis, modeling, trade studies results, or other data are too detailed to include in the body of this document, include the information as appendices.

ACRONYMS

List and define all acronyms used in the Logical Design Document.

REFERENCES

List all cited references.