Table of Contents

Introduction

Overview

This volume of procedures, standards, and forms is an important part of the documentation set describing the Nuclear Regulatory Commission's (NRC's) System Development and Life-Cycle Management (SDLCM) Methodology.

The SDLCM Methodology implements the management policy described in NRC Management Directive 2.5, "Application Systems Life-Cycle Management." NRC's SDLCM Methodology applies to, and must be followed by, all NRC employees and contractors involved in any aspect of application system development or life-cycle management.

The *SDLCM Methodology Handbook,* which is a companion document to this volume, provides the primary description of the SDLCM Methodology, its seven components, and the activities that must be performed within each of the components.

Using Procedures, Standards, and Forms

Each of the seven components of the methodology comprises a collection of *activities*. As suggested by Figure 1, the activities are not necessarily performed sequentially. Some of the activities may be performed in parallel as illustrated by the two separate paths branching from the activity shown at the top of the figure. The activities in the left branch illustrate that some iteration may be required. The activity on the lower right of the figure is performed optionally depending on some characteristic of a particular project.

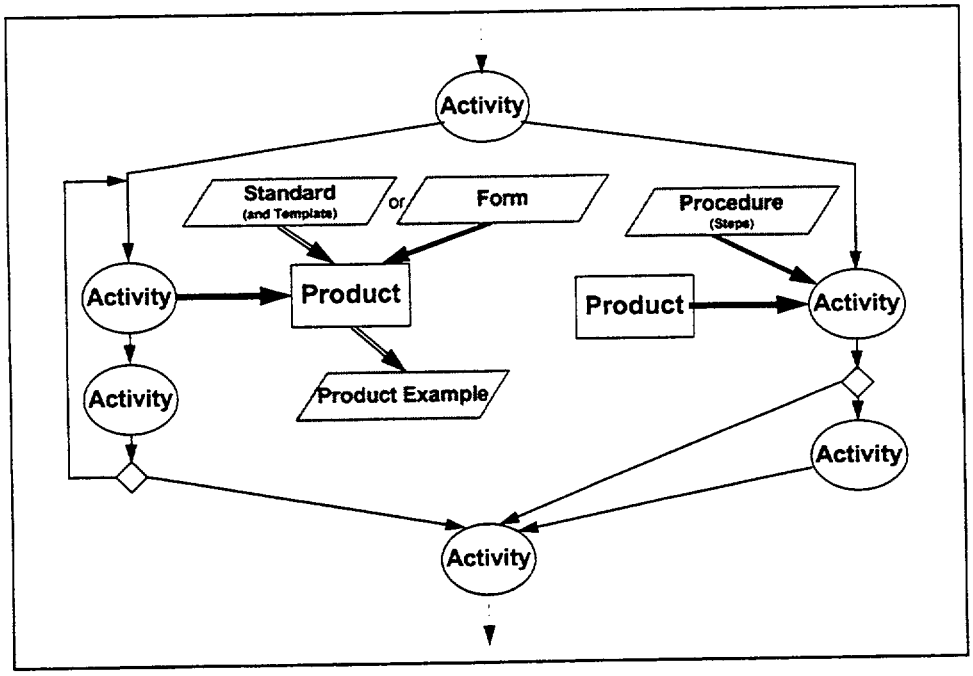

Figure 1. Relationship between Activities and Products

A *procedure* is a written description of the roles, responsibilities, and steps required for performing a complex activity or a subset of an activity. Within a procedure, *steps* are performed sequentially. Each SDLCM Methodology procedure is identified by the prefix "P–" followed by a four digit number. If an activity is described completely in the *SDLCM Methodology Handbook* or in a companion guidebook, then a corresponding procedure may not be provided.

Procedures provide the details of a process. They answer questions such as:

- "What are the steps for performing unit test?"
- "How does one conduct a peer review?"
- "What does a configuration control board do?"

A *product* is software or associated information created, modified, or incorporated to satisfy the project requirements. Examples include plans, requirements, design, code, databases, test information, and manuals. A product is an output of an activity and may be input to a subsequent activity.

A *standard* is a written set of criteria used to develop and evaluate a product or to provide and evaluate a service. Standards answer questions such as:

- "What are the form and content of a Project Charter?"
- "What is a context diagram?"
- "What is included in a software engineering notebook?"

The standards in this volume fall into two categories. A product standard (for example, the standard for a Project Action Plan) includes an annotated outline of the product. A non-product standard (for example, the standard for data models) documents a common form or approach (data models, for example, are required in several product standards). For some product standards, a word processor *template* is provided to facilitate the production of a product. Each SDLCM Methodology standard is identified by the prefix "S–" followed by a four digit number.

A *form*, rather than a product standard, is provided when the resulting product can be produced simply by filling in a set of blanks or completing a set of questions. Each SDLCM Methodology form is identified by the prefix "F–" followed by a four digit number.

A *product example* is an instance of a product developed by members of a project team using a standard or form. Representative products are provided as examples to make it easier for future project teams to satisfy the product requirements.

The specific project products required by this methodology are identified in the *SDLCM Methodology Handbook*. Product examples are maintained in the system documentation library.

Why are Procedures and Standards Needed?

Adherence to procedures and standards yields improved product quality, higher productivity, and portability of personnel. Product quality improves because everybody is using the best approach known at the present time. Productivity improves simply because nobody has to waste time deciding what to put in a document outline or developing an inspection form. Personnel become

portable from one project to another because all projects within the NRC follow the same standards.

Tailoring

The procedures, standards, and forms are designed to support the development and life-cycle management of application systems of all sizes and complexity. Consequently, some of the standards, for example, may offer more support than is required for a development effort of small size or short duration. Conversely, some may not support the more complex needs of a major development effort.

This methodology is expected to be tailored to suit the specific needs of each project. The *SDLCM Methodology Handbook* discusses tailoring. Form F–2010 may be used to request a deviation or waiver from any requirement that does not apply to a specific project.

Contents of This Volume

For the convenience of the users of the SDLCM Methodology, the procedures, standards, and forms have been numbered and grouped into related management and technical process areas as defined by SDLCM Methodology Standard S–9055 (SDLCM Methodology Document Numbering) and reflected by the Table of Contents. Experience has proved that such an ordering makes the procedures, standards, and forms easier to locate and apply. Most standards are used in more than one component of the methodology; hence, the ordering by process area tends to be more logical and easier for project team members to identify.

The companion *SDLCM Methodology Handbook* describes the use of the procedures, standards, and forms in the context of the seven components of the methodology.

Program and Project Management
Series 1000

Subject	Type	Standard
Project Charter	Identifier	S–1051
	Effective Date	October 1997
	Revision No.	

Approval C E Fitzgerald
CISSCO Program Director

A. PURPOSE

This standard specifies the content and format requirements for a Project Charter.

B. APPLICABILITY

This standard applies to all projects developing or integrating systems that enter the SDLCM Methodology at Component 1, Define Initial Project Requirements.

It is used by those persons who develop and review project charters.

C. REFERENCE PUBLICATIONS

The following publications contain related information:

- *SDLCM Methodology Handbook*, Component 1, Define Initial Project Requirements
- *SDLCM Methodology Handbook*, Appendix B, Roles and Responsibilities
- Project Initiation Plan (if the project is a result of Enterprise Integration/Migration activities
- *Enterprise Integration/Migration Strategy*
- *IRM Capital Planning and Investment Control (CPIC) Requirements*

D. STANDARD

The project charter is a management document used to initiate an NRC project that will be developed under the CISSCO Program using the SDLCM Methodology. The project charter identifies the high-level goals and objectives of the project. It specifies the project's key personnel, including the Executive Sponsor and Overall Project Manager, and commits their time and resources. It provides the background, scope, and a high-level approach for its development. It also identifies any constraints and critical success factors necessary to the management of the project.

The following paragraphs describe the content of each section of the Project Charter. Delete those that do not apply, and add others to cover special topics.

1. INTRODUCTION

Provide the project's complete name and its acronym, if applicable.

1.1 Background Summary

Describe the background for the project from the client's perspective. Explain why the project is being initiated, its importance to the client, the specific results or long-term objectives that are desired, and this project's contribution to the overall strategy. Reference the project's approval or authorization citation, such as a Capital Planning and Investment Control (CPIC) requirement or a dated Staff Requirements Memorandum. Include any other information that is necessary for understanding the background of the project.

1.2 Project Objectives

Describe the project's overall goal or purpose from the customer's perspective. Include information on the business problem to be solved or the product to be developed by the project.

List the specific objectives that the project will complete to fulfill the overall goal or purpose. Use active verbs such as, analyze, develop, test, and produce, in describing each objective.

1.3 Scope

Describe the CISSCO Program interface with the customer organization. Identify any management agreements on which the success of the project is based.

Specify the boundaries of the project, including its target users; interfaces (that is, the business areas that will be affected by the project and any systems or databases to which the project will extract or provide information); and resource support for staffing, technology (for example, computers, communications), and facilities.

Specify any constraints on the project, such as, the funding available, the project duration, support organizations, and any outstanding issues or to-be-determined items. Include CPIC material summary as applicable.

2. PROJECT ROLES AND RESPONSIBILITIES

2.1 Executive Sponsor

Identify by name and organization, the project's Executive Sponsor, the person with overall responsibility for the project. The Executive Sponsor's role and general responsibilities are discussed in the *SDLCM Methodology Handbook.* Briefly describe the specific responsibilities for this project.

2.2 Overall Project Manager

Identify by name and organization, the Overall Project Manager, the person with overall responsibility for planning the project and managing the project execution on a day-to-day basis. The Overall Project Manager's role and general responsibilities are discussed in the *SDLCM Methodology Handbook*. Briefly describe the specific responsibilities for this project.

2.3 Other Key Project Personnel and Stakeholders

Identify by name and organization, any other key project personnel and stakeholders, for example: Business Advocate, Business or Technical Project Manager, key users in the customer's organization, and Development Team members such as the quality assurance manager, system engineer, or system architect.

Discuss the time and resources commitment of each key participant.

Briefly describe any project-specific major responsibilities of the other key personnel.

3. PROJECT APPROACH

Describe, at a high level, the project management and technical approaches. Identify specific policies and directives that apply to the project. Also cite any applicable laws and regulations. Include the requirement to follow the SDLCM Methodology. If known at this early date, identify the delivery method or the life-cycle model that will be used in developing the project.

4. CRITICAL SUCCESS FACTORS

Identify the factors that will have the greatest effect on the successful completion of the project. Include items that could pose risks to the project, project deliverables, and performance measures. Include CPIC material summary as applicable.

5. APPROVAL SIGNATURES

The Executive Sponsor and Overall Project Manager sign and date the project charter to indicate their agreement. The Executive Sponsor may indicate other key personnel as additional signatories.

Subject	Type	Standard
Project Action Plan	Identifier	S–1052
	Effective Date	October 1997
	Revision No.	

Approval _____
CISSCO Program Director

A. PURPOSE

This standard specifies the content and format requirements for a Project Action Plan (PAP).

B. APPLICABILITY

A PAP is prepared for all projects that are developed using the SDLCM Methodology. Part 1, the project management plan, is required for all applicable projects; Part 2, the software development plan, is required for applicable projects that include software development or integration.

The Overall Project Manager is responsible for developing and maintaining the PAP, key managers and quality assurance personnel for reviewing it, and the Executive Sponsor for approving it. The plan is made available to all members of the project team preferably in electronic form.

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
- *SDLCM Methodology Handbook*, Component 3, Outputs and Deliverables
- SDLCM Methodology Standard S-1051, Project Charter
- SDLCM Methodology Standard S-3051, Project Definition and Analysis Document
- SDLCM Methodology Standard S-3162, Context Diagrams
- SDLCM Methodology Standard S-3151, Data Models

D. STANDARD

The Project Action Plan (PAP) communicates to CISSCO management, the customer, and project members the overall plan for performing and managing the project from start to end. It is a working document that is created initially as an activity within Component 1 of the SDLCM Methodology. The PAP grows and is updated at defined milestones during the project's life cycle and during the process itself as the project matures.

The PAP consists of two parts:

- Project management plan
- Software development plan

Part 1, the project management plan, defines the activities to be accomplished to satisfy project requirements, provides a master schedule and staffing plan for the performance of these activities, and describes the management and technical approach to accomplishing the project objectives. This part of the PAP is created as an activity within Component 1 of the SDLCM Methodology and is updated as the project grows and matures.

Part 2, the software development plan, provides 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 developed on projects that include software development or integration after project requirements have been identified as an activity within Component 3 of the SDLCM Methodology. The software development plan portion of the PAP also grows and is updated as the project progresses and matures.

The level of detail and content of the PAP 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. Delete those subsections that do not apply, and add others to cover special topics. Include a level of detail that allows for successful management of the project. Some examples of tailoring this standard include:

- The PAP for a project with a schedule of only one month, may be as short as two pages.
- The PAP for a project that is authorized for Component 1 activities may include only sufficient detail to manage those activities. If additional work is authorized following the Component 1 activities, the PAP will be updated to include the remainder of the project's activities.

Some of the information contained in the PAP is based on information developed for the Project Charter and the Project Definition and Analysis (PDA) Document. 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 PAP.

1. INTRODUCTION

Briefly describe the purpose of the project. Using the Project Charter as a starting point, identify the applicable project, define the scope of the PAP, specify the applicable references to other project documents (such as the Project Charter, PDA Document), 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 project is being initiated, its importance to the client, the specific results or long-term objectives that are desired, and this project's contribution to the overall strategy. Include the problem to be solved or the product to be developed in sufficient detail to support planning.

Refer to the Project Charter or PDA Document as appropriate for background material. If necessary, copy important information from these documents and build on it to clarify or emphasize aspects applicable to this plan.

1.2 Objectives

Specify the objectives that this project is to support from the customer's perspective, the critical functions the project should achieve, and the quantifiable criteria the project must meet to succeed. If the PAP is for a new system or service, identify any high-level requirements or, if applicable, reference an existing document (such as the PDA Document) that contains the requirements.

Refer to the project objectives contained in the Project Charter or PDA Document as appropriate. If necessary, copy important information from these documents and build on it to clarify or emphasize aspects applicable to this plan.

1.3 Scope

Define the customers for whom the work is being done and describe the CISSCO Program interface with the customer. Describe any management agreements on which the success of the plan is based.

Describe any external influences and impacts, such as all organizations involved in implementing the PAP, data and system interfaces, customer or other business needs, and regulatory requirements that are addressed by the project.

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

Refer to the project scope contained in the Project Charter or PDA Document as appropriate. If necessary, copy important information from these documents and build on it to clarify or emphasize aspects applicable to this plan.

1.4 Assumptions

Specify the major assumptions used to establish the project estimates, plans, and approach. Note: Assumptions that apply to the PAP are items that form the basis for project decisions, but have not been explicitly stated as requirements or background to the project.

Include each identified open issue if the project 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 or PDA Document 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 creation of the PAP, to provide additional information related to the project, or to be used in completing 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 any documents actually cited in the text should be listed in the references section at the end of the document.

1.6 Definition of Terms

List the terms and definitions specific to the project planning process and the project.

1.7 Overview

Discuss the organization of the PAP. 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 PAP will be updated.

2. PART 1—PROJECT MANAGEMENT PLAN

Part 1 of the PAP, the project management plan, communicates to CISSCO Program management, the business area customers, and project members the overall plan for performing and managing the project from start to end. The project management plan defines the activities to be accomplished to satisfy project requirements, provides a master schedule and staffing plan for the performance of these activities, and describes the management and technical approach to

accomplishing the project objectives. This part of the PAP is created as an activity within Component 1 of the SDLCM Methodology.

Part 1 is divided into five major sections:

1. Project definition
2. Project performance plan
3. Risk management
4. Quality assurance, configuration management, data management, and records management
5. SDLCM Methodology tailoring

Tailor the following paragraphs to be consistent with the size, scope, and complexity of the project. Delete those subsections that do not apply, and add others to cover special topics. Include a level of detail that allows for successful management of the project.

2.1 Project Definition

Document the agreement between the Overall Project Manager and the Executive Sponsor about the breadth and depth of the work required to complete the project.

If applicable, paraphrase information included in the PDA Document that defines this agreement or reference the documents that contain the appropriate information.

Define the work to be performed. If applicable, describe the work in the same terms as any related project(s).

Identify the high-level system(s), both existing and proposed, included in the project. Refer to the PDA Document for the context diagram and any other high-level diagrams needed to show the relationship of components of the systems included in the project.

Provide background and technical summary information so that the overall scope of the project is understandable.

2.1.1 DELIVERABLES

Identify and briefly describe the major work products whose satisfactory delivery is necessary for project completion.

2.1.2 MANAGEMENT APPROACH

Describe the management approach for ensuring that project commitments are met. Define the rationale for this approach. Identify the applicable policies, directives, procedures, and standards to be employed on the project.

2.1.3 TECHNICAL APPROACH

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

2.2 Project Performance Plan

Describe the framework for managing and integrating the project and specify the overall schedule and organization critical to the project's success.

2.2.1 WORK BREAKDOWN STRUCTURE

Include a Work Breakdown Structure (WBS). Describe how the project has been decomposed into major units of work. Include all components of the solution and identify those that have been authorized for work.

Include a description of the units of work (or activities) and the deliverables that one or more persons are to complete within a given period of time.

2.2.2 PROJECT SCHEDULE

Include both parts of the project schedule as applicable. The Gantt or milestone charts show the timing of all components within the project; and the task assignment schedules show how unit of work are assigned to team members.

If appropriate, include a network diagram that shows all key interdependencies or critical paths.

2.2.2.1 Gantt or Milestone Charts

Include Gantt or milestone charts (or similar diagrams) to show the start and finish dates, as well as the duration of the development and delivery of all major components of the project . The top level is a one-page chart that shows the approximate start and end dates of each major component over the anticipated duration of the project. The detailed level shows the major units of work that make up the project.

2.2.2.2 Task Assignment Schedules

Include task assignment schedules to show how units of work identified in the detailed schedule have been assigned to team members. Use task assignment schedules to show each major activity or unit of work, its budgeted time for completion, and its planned start and end dates.

2.2.3 PROJECT ORGANIZATION

Describe the organization of the project that will implement the PAP. Include an organization chart of the project team.

2.2.3.1 Roles and Responsibilities

Describe the team members' responsibilities, accountabilities, and reporting relationships. Use standard (SDLCM Methodology) role descriptions and responsibilities.

2.2.3.2 Skills

Identify any special skills needed to perform the work described in this PAP. Address how these skills are being met (for example, assignment of skilled personnel, training).

2.2.3.3 Interfaces

Describe the interfaces, both management and technical, between the project team and the customer (business area) organization.

Identify any steering committees, working groups, or boards that will be used to help govern or to provide for technical interchange during project implementation.

2.2.3.4 Staffing Profile

Include the staffing profile proposed for the project. The staffing profile should identify the number and type of personnel needed for the project, broken down by role and time period.

2.3 Risk Management

Describe the overall activities that the project will undertake to reduce risk. Describe the project's approach to risk management.

2.3.1 RISK DESCRIPTION

Describe the project's level of risk. Identify major project risks, their areas of impact, and their potential effect on project success. Describe the risks from the perspective of both customer and project team.

2.3.2 RISK MANAGEMENT

Describe the process established to manage risks, including the monitoring of risks and identification of activities to reduce the occurrence of specific risks. Specify the level of involvement of business area customer, Information Resources Management (IRM) personnel, and contractor personnel in managing the identified risk factors.

2.3.3 RISK MITIGATION

Describe the process established to mitigate risks. Include the development of appropriate risk mitigation plans for each risk identified.

2.4 Quality Assurance, Configuration Management, Data Management, and Records Management

Describe quality assurance (QA), configuration management (CM), data management (DM), and records management as they apply to the project.

2.4.1 QUALITY ASSURANCE

Describe the approach to quality assurance that ensures that all development products (documentation, hardware, software, and data) meet project requirements and applicable standards. Describe the organization, methods, and standards used to ensure the quality of the project's processes and products.

Explicitly identify the QA activities that will be performed on the project and the schedule for their performance. The schedule may be provided separately in this section or it may be included as part of the overall project schedule in Section 2.2.2.1.

Reference the CISSCO Program Quality Management Plan as appropriate. If a project-level QA plan has been developed, reference that plan.

2.4.2 CONFIGURATION MANAGEMENT

Describe the configuration management approach for the project. Identify the project baselines to be established. Identify the project-specific documents and project databases that will be used to accomplish configuration management. Identify the person(s) responsible for CM and describe their roles. Describe the tools and procedures used to ensure the integrity of the system configuration; how changes are requested from the baseline and how such changes are approved and implemented, and who makes the changes.

Explicitly identify the CM activities that will be performed on the project and the schedule for their performance. The schedule may be provided separately in this section or it may be included as part of the overall project schedule in Section 2.2.2.1.

If a project-level CM plan has been developed, reference that plan.

2.4.3 DATA MANAGEMENT

Describe the process for creating, accessing, copying, and maintaining project documents and other project-related data. Identify the person(s) responsible for DM and describe their roles. Specify the documents and data to be managed, and their storage locations; and describe rules about retaining project documents once the project is completed.

Explicitly identify the DM activities that will be performed on the project and the schedule for their performance. The schedule may be provided separately in this section or it may be included as part of the overall project schedule in Section 2.2.2.1.

2.4.4 RECORDS MANAGEMENT

Describe the records management approach for the project. Specify the records to be managed, and their storage locations; and describe the records management rules for retention once the project is completed. Reference the appropriate records management procedures and standards that will be followed on the project. Identify a specific point of contact for coordination within the Records Management Branch.

2.5 SDLCM Methodology Tailoring

Specify any tailoring of the SDLCM Methodology applicable to the project. The tailoring may be done incrementally as approval is granted to perform work in each component area of the methodology. Include the checklist for new projects from the *SDLCM Methodology Handbook* as an appendix to this plan to indicate at the component and activity levels the tailoring that has taken place.

Identify any exceptions, and approvals of any exceptions, to NRC and SDLCM Methodology policies, directives, procedures, and standards that apply to this project.

3. PART 2—SOFTWARE DEVELOPMENT PLAN

Part 2 of the PAP, the software development plan, defines the project's approach to software development and integration for the project. It describes the approach to managing software development and integration for the project, 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 or integration after project requirements have been identified as an activity within Component 3 of the SDLCM Methodology.

Tailor the following paragraphs to be consistent with the size, scope, and complexity of the project. Delete those subsections that do not apply, and add others to cover special topics. Include a level of detail that allows for successful management of the software development and integration activities of the project.

3.1 Overview of Software Development and Integration Activities

Consistent with the software life-cycle model identified in Section 2.1.3, 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.

3.2 Software Development Organization and Responsibilities

If the project team is relatively small and the software team organization was adequately addressed in Part 1, Project Management Plan, then this section does not apply. If the software team is a part of a larger project team described in Part 1, then describe the software team in this section.

3.2.1 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 member's authorities and responsibilities.

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

3.2.2 INTERFACING GROUPS

Identify interfacing groups, points of contact, and the responsibilities of each group. Use a table or diagram to illustrate these interfaces.

3.3 Software Development Technical Approach

If the project is (almost) exclusively software development, and the technical approach was described in Part 1, Project Management Plan, then this section may not apply. If this section is included, then reference any previously developed documentation, such as the Project Management Plan section of this document or the PDA Document, as applicable.

3.3.1 ACTIVITIES, TOOLS, AND PRODUCTS

For the project's selected life-cycle model, identify the major activities to be performed, the development methods and tools that will be applied, and the products that will be developed during each life-cycle phase.

3.3.2 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.

3.3.3 INTEGRATION AND TESTING

Describe the process for integrating the builds and releases.

Define the scope and hierarchy for testing. State the level of independence of the various test teams. Summarize any critical high-level acceptance testing criteria.

3.4 Software Development Management Approach

Describe items that affect the management approach, including project priorities

3.4.1 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), by software component (subsystem, object, or unit)
- Training requirements
- Computer resources

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

3.4.2 SOFTWARE DEVELOPMENT MILESTONES AND SCHEDULES

Specify the work to be done, who will perform it, 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)

Use tables if appropriate.

3.4.3 SOFTWARE DEVELOPMENT MEASURES

Identify the indicators that management will use to monitor progress and product quality. Measures, such as effort in staff months, duration in months, pages of documentation produced, annual maintenance cost, average staff size, 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.

ACRONYMS

List and define all acronyms used in the Project Action Plan.

REFERENCES

List all cited references.

Subject	Type	Standard
Statement of Work	Identifier	S–1053
	Effective Date	October 1997
	Revision No.	

Approval _____
CISSCO Program Director

A. PURPOSE

This standard specifies the content and format requirements for the Statement of Work (SOW) document.

B. APPLICABILITY

This standard applies to all NRC projects, subject to the SDLCM Methodology, that plan to contract, via CISSCO or other contract vehicle, for any necessary resources (for example, staff, technology, equipment, training) to support the project.

C. REFERENCE PUBLICATIONS

The following publications contain related information:

- *SDLCM Methodology Handbook*, Component 2

D. STANDARD

An SOW is the project manager's description of the resources required from a contractor who will provide them. It serves as the basis of a contractual agreement with the supplier for the acquisition of resources required for the project.

Any number of Statement(s) of Work may be generated as an output of an activity of Component 2, Acquire Support Resources, of the SDLCM Methodology. The following examples illustrate the format (embedded in another document or stand alone), preparer (project manager or CISSCO contractor), and recipient (CISSCO contractor, subcontractor, or vendor) of an SOW:

- An SOW may be prepared by the project manager and embedded in a TAC assigned to the CISSCO contractor to encompass the work required for any Component of the SDLCM Methodology.
- An SOW may be written by the CISSCO contractor as a deliverable of Component 2 activities to identify the work required during future SDLCM Methodology Components applicable to the project.
- An SOW may be included as the definition of the work to be performed or the resources to be provided by a CISSCO subcontractor, other contractor, or vendor.

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

1. INTRODUCTION

1.1. Background

Describe the client's business and identify how the requested resources will fit into the overall project.

1.2. Objectives

Specify the project objectives and critical functions the requested resources should achieve.

Include the specific requirements for the resources. Based on the overall project schedule, identify the milestones that must be met by the resource provider.

1.3. Scope

Identifies the aspects of the project, such as customers, products, processes, organizations, locations, or applications, to be included in this acquisition of resources and those to be excluded. Identify other external influences and impacts, such as interfaces, customer needs, and regulatory requirements, to be addressed. Describe the work activities to be completed and the work products to be delivered.

2. APPROACH

Specify the approach required, if applicable, to achieve the results and any project management standards that will be used to control and monitor the acquisition of the resources.

3. CLIENT RESPONSIBILITIES

Identify the client's (that is, the project's) responsibilities in acquiring the requested resources. These responsibilities could include such items as equipment or software specifications or skills required of personnel resources. Note: The project must understand not only its responsibilities but also the impact on the supplier of the resources and the project if these responsibilities are not fulfilled.

4. COMPLETION CRITERIA

Identify the major work products that must be accepted by the project before the contract for the requested resources can be considered complete.

In the case of equipment (hardware or software) or training, include any necessary documentation as work products.

In the case of requests for staffing to perform specific function, include any status reports or reviews as work products.

Subject	Type	Standard
Conversion Plan	Identifier	S–1054
	Effective Date	October 1997
	Revision No.	

Approval C E Fitzgerald
CISSCO Program Director

A. PURPOSE

This standard specifies the content and format requirements for a Conversion Plan.

B. APPLICABILITY

This standard applies to all NRC projects, subject to the SDLCM Methodology, that plan to develop and deploy a new, upgraded, or migrated data processing system or software application.

This standard is used by those persons who create and review the Conversion Plan.

C. REFERENCE PUBLICATIONS

The following publications contain related information:

- *SDLCM Methodology Handbook*, Component 3
- SDLCM Methodology Standard S–5051, Tactical Integration Plan
- *NRC Model Tactical Integration Plan*, January 3, 1997

D. STANDARD

The Conversion Plan provides a detailed description of what data will be converted, where the data will come from, and how the supporting business process will be organized, staffed, and scheduled.

Tailor this standard as needed to be consistent with the size, scope, and complexity of the conversion effort. Several sections have been marked optional and are included only if the identified conversion activities are required. Add sections and subsections for special topics and delete sections and subsections that are not applicable. For projects with very small conversion efforts, the Conversion Plan may be included in its entirety as Section 2.5 of the Tactical Integration Plan.

The following paragraphs describe the content of each section of the Conversion Plan.

1. INTRODUCTION

Describe the purpose of the Conversion Plan.

1.1 Background

Provide any information necessary for understanding the project's data conversion effort.

1.2 Objectives

Include project objectives related to data conversion. Omit this section if such objectives have not explicitly stated in prior project documentation.

1.3 Scope

Describe the scope of the conversion effort, including, but not limited to, whether it will be formal or informal, automated or manual or a combination of the two, and the corporate or legacy system databases that are included. Identify any constraints associated with the conversion effort.

1.4 Assumptions

Include any assumptions related to data conversion.

1.5 Applicable Documents

Provide a list of reference documents for use by personnel involved in the preparation and accomplishment of data conversion for the project. Include applicable standards and previously developed system and project documentation, such as:

- Project's Project Action Plan
- Project's Definition and Analysis Document
- Project's Tactical Integration Plan
- Project's Solution Logical Design Document

2. DATA CONVERSION APPROACH

Provide a data conversion plan overview consistent with the application data model. Include descriptions of the applicable data conversion considerations. The following are examples of data conversion considerations:

- Identify databases, repositories, and hardware devices that exchange data with the application or system.

- Describe how exchanged data will be examined to ensure that all software applications use the exact same format, and, if not, the process for negotiating with the owner of the data for a means to achieve synchronization.
- Discuss how any change in database software, schema, or software version will be accommodated by conversion prior to the operational date.
- Discuss the impact of geographically distributed databases on the conversion process.
- Determine the need for any conversion-specific testing and for developing, if necessary, a test database to be used in development or acceptance testing.

3. DATA CATEGORIES

List major categories of data to be converted. Group closely-related data entities into one category.

3.1 Data Category 1

Describe the conversion approach or approaches to be used for data category 1. Note: The main approaches are automated conversion, manual entry, and automated conversion with manual review and correction. If more than one approach is to be used for a data category, indicate the approximate percentage split among the conversion approaches.

Include a Data Flow Diagram to describe the flow of work associated with manual conversion or correction activities.

3.*n* Data Category *n*

Provide the same information for each additional data category as for data category 1.

4. FORMAL DATA CONVERSION (OPTIONAL)

If a formal data conversion effort is required, include the following information, as applicable.

4.1 Data Conversion Procedures

Specify the schedule and identify the author(s) who will develop the following detailed conversion procedures:

- Acquiring the data to be converted
- Validating converted data
- Synchronizing converted data with the source legacy data
- Transition to use of the converted data
- Documenting the converted data formats, schema, etc.

4.2 Conversion Tools

Identify any tools that will be used to support the data conversion process.

4.3 Effects on Other Systems

Describe the potential effects of the data conversion process on other interfacing systems. If any of these effects are negative, identify the intended resolution.

5. CONVERSION SCHEDULE AND STAFFING

5.1 Conversion Schedule

Provide a conversion schedule (Gantt Chart), relative to the initial operating date, that covers each database or type of data to be converted and shows the approximate time involved to convert each.

5.2 Conversion Staffing (Optional)

If manual conversion activities are included, provide staffing estimates. Include the following information for each different conversion role:

- Name of role
- Number of people required
- Training or skill required
- Probable source (hire, in-house, transfer, or temporary staff)
- Duration of requirement

6. BACKUP STRATEGY

Document the project's backup or reversion strategy in the event that the data conversion is impaired or not completed due to unforeseen circumstances.

ACRONYMS

List and define all acronyms used in the Conversion Plan.

REFERENCES

List all cited references.

Subject	Type	Standard
Development and Maintenance Environment Products Installation Plan	Identifier	S–1055
	Effective Date	October 1997
	Revision No.	

Approval C E Fitzgerald
CISSCO Program Director

A. PURPOSE

This standard specifies content and format requirements for the SDLCM Methodology Development and Maintenance Environment Products Installation Plan.

B. APPLICABILITY

This standard applies to all projects that fall under the SDLCM Methodology umbrella and that require the installation of hardware or software as part of their Acquire Support Resources component. It is used by those persons who plan for the acquisition of resources for subsequent design and development components.

Whereas the Tactical Integration Plan specifies an installation plan for deployment, this Development and Maintenance Environment Products Installation Plan applies specifically to the installation of products for the development environment.

C. REFERENCE PUBLICATIONS

The following publications contain related information:

- *SDLCM Methodology Handbook*

D. STANDARD

The Development and Maintenance Environment Products Installation Plan provides a detailed description of the activities involved in the installation of equipment resources required for the Design the Solution component. It defines responsibilities, schedules, risks, and risk mitigation approaches.

The standard outline for this document is discussed below. Enhance the outline as needed by adding subsections and separately numbered items to the sections in the standard outline. Add sections and subsections for special topics. Sections and subsections that are not applicable should *not* be deleted; they should indicate "not applicable."

1. INTRODUCTION

1.1 Background

Describe the need for the installation of support resources.

1.2 Objectives

Describe the capability that will exist after the installation of the support resources.

1.3 Scope

Define the Development and Maintenance Environment Products Installation Plan and what it describes, such as hardware and software installation, for a given project. Describe the activities that mark the beginning and end of the installation. Also describe what it does not cover.

1.4 Assumptions

Discuss what we have assumed because something we need to know has not been explicitly stated as a requirement or background.

1.5 Applicable Documents

List the documents that support this plan or that provide additional information about the various activities included under this plan, such as hardware and software installation and maintenance manuals and procedures; test requirements; training materials; applicable system requirements and specification documents; project or mission schedules; facility site requirements, plans, and activity schedules; and problem resolution procedures and policies.

2. APPROACH

Describe the various stages of activity and the strategy for accomplishing the installation and turnover. Describe the organizations involved and their responsibilities. Describe how coordination will be maintained among these organizations. Clearly identify the date or activity at which turnover occurs.

3. GENERAL SYSTEM INSTALLATION

Describe the organization receiving the system and the facility that houses it.

3.1 System Overview

Summarize the system's function and the context in which it operates, including the interfaces. State any special conditions that exist and how they might affect this plan or its implementation.

State whether this is an entirely new system or just new hardware or new software. If this is not a new system, describe the old one and what components or functions are being replaced or modified. If software maintenance releases exist, describe the releases and what they are intended to accomplish, such as resolution of test discrepancies, software enhancement, a phased improvement plan, or patches for operational problems. State whether the new software or hardware will be operated in the environment in which it is acceptance tested or broken down and rebuilt in some other location.

3.2 Installation Considerations

State the considerations that must be met during system installation. The following considerations are typical of what needs to be addressed:

- Hardware or software to be installed
- State of readiness required at installation site
- Physical, security, and safety restrictions for hardware, software, or personnel entering the facility
- Preshipment checkout of hardware, software, or data
- Special shipping, packaging, or delivery needs
- Special communication needs
- Configuration and data management needs
- Engineering support needs
- How and where acceptance testing is to be performed
- Hardware and software acceptance criteria
- Certifications or inspections to be performed
- Activities sequences that must be adhered to
- Simulation requirements
- Staff training needs for acceptance testing, operations, and maintenance
- Required date by which turnover must be complete
- Problem reporting and tracking

3.3 Installation Milestones

Prepare the milestone schedule for the installation period. Indicate the major hardware and software transitions and other associated activities, such as acceptance testing. Include site preparation milestones that can affect installation activities.

3.4 Interfaces

Describe the system's interfaces and the transitions that will be made over the course of installation. List the hardware and software configuration changes that must be accomplished to support these transitions. Describe the organizations responsible for the various interfaces.

3.5 Problem Identification and Resolution

Describe the problem reporting and tracking procedures. Identify who is responsible at various stages for the defined activities. Describe the problem resolution procedures and how fixes or changes are introduced into the installation process.

4. HARDWARE INSTALLATION AND ACCEPTANCE

If hardware installation is required, summarize what is involved: shipping and receiving, site preparation, removal of the old system, packing and unpacking, setup, startup, vendor or third-party system software installation, network connection, and acceptance testing.

4.1 Hardware Installation Approach

Describe the activities necessary to fulfill the hardware installation requirements: physical installation, equipment turn on and initial checkout, operating system and vendor or third-party software installation and testing, and hardware acceptance. If this is a replacement system and the old system is being retained for the transition period, describe plans for parallel system operations and, in case of need, procedures for fallback to the old system. Consider whether it would be of use to bridge the switchover period with an alternate processor.

4.2 Hardware Installation

List the hardware being installed or retained. Describe the nominal sequence of hardware installation activities.

4.3 Hardware Acceptance

List and describe the acceptance activities associated with the new or reconfigured system. This can include both pre- and post-installation checkout and the exercising of the physical interfaces. Describe the conditions necessary for acceptance testing to begin and to be concluded successfully. Describe the acceptance criteria.

4.4 Responsibilities

Allocate responsibility for the various hardware installation, acceptance test, switchover, and fallback activities.

5. SOFTWARE INSTALLATION AND ACCEPTANCE

Summarize what is involved: vendor or third-party commercial off-the-shelf (COTS) software installation, application software installation and generation, network installation, user allocations, logical name assignments, system tuning, and acceptance testing.

5.1 Software Installation Approach

Describe the activities necessary to fulfill the software installation requirements: system configuration; software installation, setup, and testing; file allocation and user account creation; network installation; operating system and vendor or third-party COTS software installation and testing; and installation of monitoring tools. Describe procedures for the final switchover to operations and develop the initial system tuning parameter values.

As necessary, establish policies and procedures for installing and accepting new software releases, ensuring integrity and security of the system and its data, and monitoring resource usage. If this is a replacement system or a new software release and the old software is being retained for the transition period, develop plans for parallel operation of the old and new software systems and, in case of need, procedures for fallback to the old software.

5.2 Software Installation

List the software being installed or retained. Describe the nominal sequence of software installation activities.

5.3 Software Acceptance

List and describe the acceptance activities associated with the new software. These activities can include formal reviews and inspections by the customer or purchasing agency. Describe the conditions necessary for acceptance testing to begin and to be concluded successfully. Describe the acceptance criteria or refer to the acceptance test plan.

5.4 Responsibilities

Allocate to specific individuals the responsibility for the various software installation, acceptance test, switchover, and fallback activities.

6. TRAINING REQUIREMENTS

State training requirements for personnel. List the tasks to be performed and describe the skills and knowledge necessary for accomplishing each task.

Identify the positions that are to be staffed. Describe any cross-training requirements.

7. RISKS AND RISK MITIGATION

Enumerate the risks associated with installation and turnover of hardware or software and COTS, availability of key personnel, delays in facility or site preparation, failures of hardware or software, delays in acceptance of all or part of the system, deficiencies in operational procedures, and inexperience of staff. Analyze the potential impact of these risks and describe any proposed risk mitigation strategies, such as activity resequencing, staffing changes, activity regrouping, vendor changes or COTS substitution, and equipment rental. Analyses should consider critical project or system functional objectives that may be jeopardized.

Subject	Type	Standard
Security Controls	Identifier	S–1056
	Effective Date	November 1998
	Revision No.	1

Approval C E Fitzgerald
CISSCO Program Director

A. PURPOSE

Federal agencies are required to establish controls to ensure adequate security for all information processed, transmitted, or stored in federal automated information systems. This standard specifies the minimum set of security controls to be included in all developed systems.

B. APPLICABILITY

This standard applies to

1. All NRC projects involved in the development of general support systems, under the presumption that all contain some sensitive information
2. High-risk major applications

C. REFERENCE PUBLICATIONS

The following publications contain related information:

- *SDLCM Methodology Handbook*
- Office of Management and Budget Circular A–130, Appendix III, "Security of Federal Automated Information Resources," revised February 20, 1996.
- "Instructions for Preparing Security Plans for Local Area Networks in Compliance with OMB Bulletin No. 90-08," NUREG/BR-0166, February 1992
- "User Guide for Developing and Evaluating Security Plans for Unclassified Federal Automated Information Systems," NIST, Draft, July 1997.

D. STANDARD

OMB Circular A–130, Appendix III, sets forth four security controls:

1. Assigning responsibility for security
2. Security Planning
3. Periodic review of security controls
4. Management authorization

The appendix requires that these controls be applied in two areas of management responsibility:

1. General support systems

2. Major applications

These controls must be included as an integral part of the systems development and life-cycle management process for both general support systems and major applications.

For further information on complying with this standard, refer to the reference publications.

Subject	Type	Procedure
Status Tracking and Reporting	Identifier	P–1101
	Effective Date	October 1997
	Revision No.	

Approval C E Fitzgerald
CISSCO Program Director

1. PURPOSE

This procedure provides direction on how to track and report project status. It also establishes a standard approach for tracking cost and schedule performance throughout the life of the project. By using an objective measure of progress, Earned Value, an accurate indication of the status of the project can be determined and cost at completion can be estimated.

2. APPLICABILITY

Tracking and reporting of project status is required of all projects subject to the SDLCM Methodology.

3. REFERENCE PUBLICATIONS

The following publications contain related information:

- *SDLCM Methodology Handbook*
- SDLCM Methodology Procedure, P–2101, Peer Review

4. PROCEDURE

4.1 Data Flow Diagram

The project status tracking and reporting procedure comprises the eight steps identified in the data flow diagram shown in Figure 1101–1.

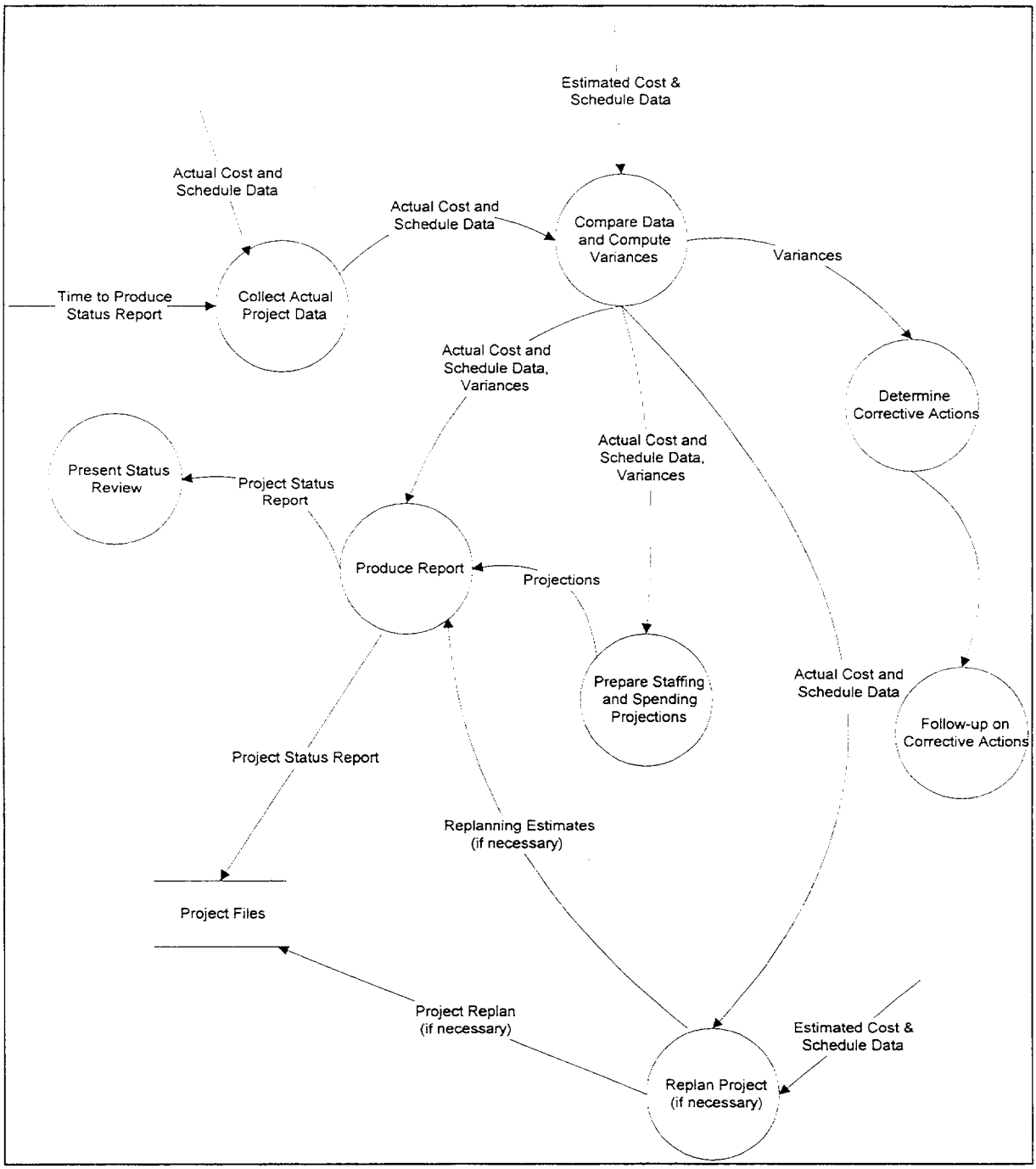

Figure 1101–1. Status Tracking and Reporting

4.2 Entry Criteria

Use this procedure monthly, or as scheduled, for project status reporting. Inputs include the baseline estimates for the project and the actual schedule, staffing, and cost data for the reporting period.

4.3 Steps

The following steps comprise the status tracking and reporting procedure.

1. Collect the actual schedule, staffing, and cost data for the reporting period. [Note: This information may be accumulated in a contractor's project tracking system or in the project's C/S 10,000 records.]
2. Compare the actual data with that planned for the work period and compute variances in cost or schedule.
3. Determine any corrective actions that need to be taken to reduce either cost or schedule variances. Document them.
4. Prepare staffing and spending projections for the remainder of the authorized project period.
5. Produce the status report in the format required for the project. Obtain quality assurance (QA) review of the report.
6. Deliver the written report and present the highlights as required in a scheduled status review meeting with the Business Advocate. [Note: If the status report is associated with work assigned to a contractor and is produced by the contractor, the contractor should review the status with the Overall Project Manager prior to reviewing the information with the Business Advocate.]
7. If necessary, replan the project to update the baseline estimations. [Note: This should only happen when the direction of the project has changed sufficiently to warrant it.]
8. Follow up on corrective actions during the next reporting period. Report on their status in the next scheduled status report.

4.4 Exit Criteria

The outputs of this procedure are the project status report for the period and, if necessary, an updated estimation baseline.

4.5 Verification

Submit all project status reports to QA personnel for review.

4.6 Roles

Table 1101–1 is a Step-Role Table for the project tracking and reporting procedure:

Table 1101–1. Project Tracking and Reporting Step-Role Table

Steps: / Roles:	Business Advocate	Overall PM	Tech PM	Project Leads	QA
1. Collect the actual data for the reporting period.				P	
2. Compare the actual data with that planned for the work period and compute variances in cost and schedule.			R	P	
3. Determine any corrective actions that need to be taken to reduce variances. Document them.			R	P	
4. Prepare staffing and spending projections for the remainder of the authorized project period.		R, A	P	S	
5. Produce the status report.		R, A	P	S	R
6. Present the highlights as required in a scheduled status review meeting.	R	P			
7. If necessary, replan the project to update the baseline estimations.	R, A	P	S	S	R
8. Follow up on corrective actions.		R	P	S	R

Legend: P = Performs, R = Reviews, A = Approves, S = Supports

Subject	Type	Procedure
Environment Change	Identifier	P–1601
	Effective Date	December 1999
	Revision No.	

Approval C E Fitzgerald
CISSCO Program Director

1. PURPOSE

This procedure establishes the mechanism for requesting and effecting changes to the NRC hardware and software environment for both infrastructure and applications support.

2. APPLICABILITY

This procedure applies to all requests for changes to the NRC hardware and software environment, including (but not necessarily limited to) the following elements:

- Platform hardware (server or desktop)
  - Processor
  - Communications
  - Storage
  - Memory
- Platform Software (specific version, release, and patch)
  - Operating system (desktop, server, or host)
  - Compilers and interpreters
  - Groupware
  - Database Management Systems
  - User interfaces, including images
  - Transfer Protocol Software
  - Utilities
- Application Development and Maintenance Tools
  - Vendor software (specific version and release)
  - Add-on software (specific version and release)
  - Libraries (shared code, call libraries, DLL's, etc.)
  - Reusable components

The NRC Environment Configuration Control Board (CCB) may consider requests for changes to other environmental elements not specifically itemized in the list above.

SDLCM Methodology Procedure P–2501 (Configuration Control Board) defines the activities of the NRC Environment CCB (and all other NRC CCBs). The interface between this procedure and the CCB procedure is defined herein.

Any NRC personnel and any personnel from NRC contractor organizations may submit an Environment Change Request (ECR). ECRs propose additions, deletions, or modifications to the current environment[1].

3. REFERENCE PUBLICATIONS

The following publications contain related information:

- *SDLCM Methodology Handbook*
- SDLCM Methodology Form F–1601, Environment Change Request Form
- SDLCM Methodology Procedure P–2501, Configuration Control Board
- NRC Technical Reference Model, NRC/OCIO

4. PROCEDURE

4.1 Data Flow Diagram

The Environment Change procedure has the five major steps identified in the data flow diagram shown in Figure 1601–1.

4.2 Entry Criteria

This section identifies the inputs and triggers.

The following input is necessary to begin this procedure:

- A recognized need for an addition, deletion, or modification to the NRC environment

Any of the following events may trigger this procedure:

- A failure of the environment to support a functional requirement
- An opportunity for process improvement
- The introduction of new technology
- Cessation of vendor support

[1] In a future revision to this procedure, the phrase "current environment" will be changed to "baselined operational environment."

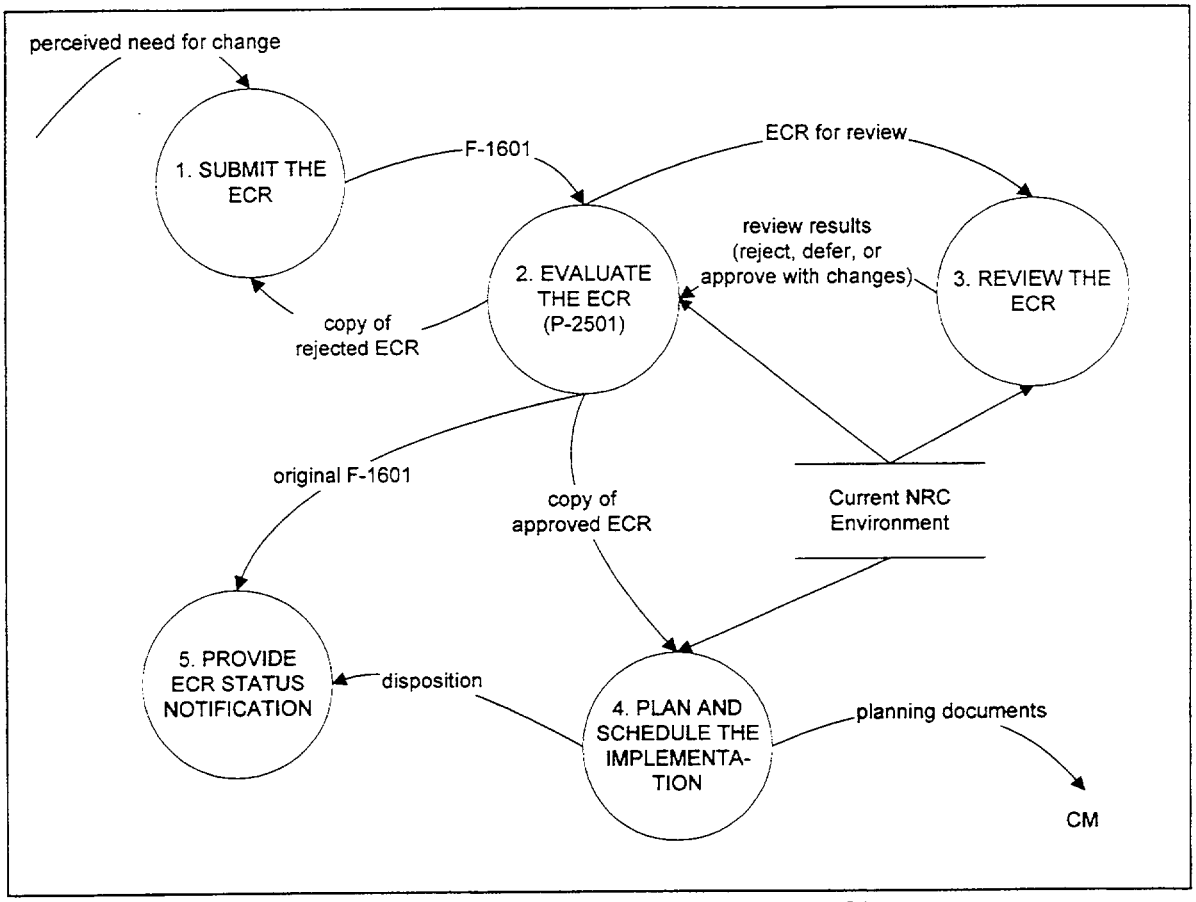

Figure 1601–1. Environment Change Data Flow Diagram

4.3 Steps

This section provides details of the steps shown in the data flow diagram (Figure 1601–1) and the data items that flow among the steps. The Step-Role table included in Section 4.6 clarifies which roles are responsible for performing which steps.

Perform the following steps:

1. Submit the Environment Change Request

   Any NRC personnel and any personnel from NRC contractor organizations may submit an ECR. ECRs propose additions, deletions, or modifications to NRC's current environment[2].

   To request a change, use SDLCM Methodology Form F–1601, Environment Change Request Form. Complete all blocks under "Originator Information" and "Change Information." Attach additional pages as needed to provide the required details, and use the form as a cover sheet to submit the request.

[2] See first footnote.

ECR Form Instructions

Originator Information

- Complete fields for Originator's Name, Organization, Location, Telephone Number, and Date Submitted.
- Provide the printed name of the Originator's Supervisor. Obtain the signature and date after completing the Change Information portion.

Change Information

- **Responsible NRC Organization.** Specify the NRC organization responsible for the environmental element to be changed.
- **Scope.** Check the applicable box.
- **Priority.** Check the applicable box. For a time-critical change, specify a deadline. For time-critical and urgent changes, provide a justification and state the effect of a delay in Part A, Items 6 and 7 (see below).
- **Brief name.** Provide a brief identification of the new hardware or software technology to facilitate references to your ECR.
- **Details.** Provide information that addresses each of the items in Parts A, B, and C, as applicable.

  Part A. Enter the following information directly on the form or provide an attachment labeled "Part A." If an item does not apply, justify why not and specify "N/A."

  1. Vendor Name
  2. Complete name of new or expanded technology, including version, release, and patch identifiers, if applicable
  3. Number of copies that will be needed
  4. Number of desktops that will be affected by the presence of either the vendor product or the output generated by the vendor product (for example, run-time modules)
  5. Source of funding to satisfy Items 3 and 4
  6. Justification for Time-Critical or Urgent priority
  7. Consequences of rejecting this request from the requestor's perspective (for example, a failure to meet an agreed-upon delivery date for a system as promised to the Commission)
  8. Compatibility with the NRC's Technical Reference Model (Preferred or Target)
  9. Required operating environment (hardware, operating system, or other products)
  10. Intended functionality that this product, upgrade, or patch will offer
  11. Summary of distinguishing features of the product, upgrade, or patch
  12. Technological benefit to the NRC
  13. Applicability of the product for other users, systems, platforms, etc. (for example, can others benefit from the change?)
  14. Availability of a competing (or comparable) technology product already in the NRC environment (for example, for a new product request, is there a similar product already installed?)

  Part B. Provide an attachment to the form labeled "Part B."

  Using the criteria listed below, provide a comparative evaluation of the requested new or upgraded technology with at least two competing products. (If the answer to Part A, Item 14 is "yes," one of these must be the currently available product.)

  For each of the criteria, rate each product numerically from 1 to 4 as follows: (1) Does not satisfy requirements, (2) Partially satisfies requirements, (3) Satisfies requirements, and (4) Exceeds requirements. Indicate not applicable (N/A) where appropriate. The first criterion (specifically required functionality) shall be weighted higher that the other criteria. Use a table or narrative as appropriate for your ECR. Provide any explanatory information needed to support your

evaluation. Conclude the comparison with a cost-benefit analysis that supports the Environment Change Request.

a. Specifically required functionality
b. Documentation support
c. Y2K compliance
d. Ease of installation, setup, and use
e. On-line help
f. Vendor training (specify costs if separate from purchase price)
g. Vendor stability
h. Product Stability
i. Licensing requirements and issues (specify costs and whether site licensing is available)
j. Upgrade to earlier version already in the environment

Part C. Optionally, provide an attachment to the form labeled "Part C" to present any other information in support of your Environment Change Request.

Submit the completed ECR form to the originator's supervisor for approval. **The supervisor submits the form to the CISSCO Configuration Management organization for processing** as defined in SDLCM Methodology Procedure P–2501, Configuration Control Board. CM processing includes reviewing the package for completeness, assigning an ECR number if complete, packaging the ECR along with other ECRs, and submission to the CCB. Review for completeness includes confirming the presence of all applicable information in Part A and a comparison of at least three products in Part B.

2. Evaluate the Environment Change Request

   The NRC Environment CCB reviews the ECR (see Procedure P–2501).

   In accordance with Procedure P–2501, the members review the ECR (or a package of ECRs) prior to the actual CCB meeting. Each CCB member's review includes, at a minimum, assessment of any effect on the member's area of responsibility. (For example, is there any effect on other environmental elements or any need for other upgrades to support the current change request?)

   The CCB shall forward the ECR to the Change Review Committee to review its potential effect on the current environment[3] (Step 3). This technical review will normally be completed prior to the scheduled CCB meeting so that the CCB members will be prepared to vote on the disposition of the ECR during the meeting.

   If the CCB rejects the ECR, the form is annotated with the justification for the rejection, and the CM organization provides a copy of the ECR form to the originator.

   If the CCB approves the ECR, CM forwards a copy to the Change Implementation Committee (Step 4).

   The CCB may also decide to defer action until a future CCB meeting date, possibly to permit additional review by the Change Review Committee (Step 3).

[3] See first footnote.

ECR Form Instructions

CCB Action

- Select one box to indicate the disposition of the ECR. Justify a rejected ECR in the Comments field. Specify a date for a deferred ECR.
- Provide the printed name and signature of the CCB Chairman. Indicate the date the ECR is signed.
- Comments. If applicable, provide comments from the CCB as feedback to the originator or for the record; attach additional pages if necessary.

3. Review the Environment Change Request

   The Change Review Committee reviews the request for its potential effect on the current environment[4]. The committee members may request additional information from the originator. The committee documents its recommendations (including any possible implementation alternatives) and returns the ECR to the CCB for further evaluation (Step 2).

4. Plan and Schedule the Implementation

   The Change Implementation Committee develops a plan, including a schedule, for implementing approved ECRs. If necessary, the committee members consult with a person or group possessing the necessary expertise or experience (possibly the originator). A copy of the reviewed and approved plan is placed under CM control.

5. Provide Environment Change Request Status Notification

   The Chairman of the Environment CCB sends an e-mail notification to all personnel within the Office of the Chief Information Officer and to the members of the Information Technology Business Council announcing the nature of the change and the schedule for implementation.

   The CM organization notifies the requester of the final disposition of the ECR.

   The CM organization maintains the status of all ECRs; ECR status is available for review by any personnel throughout the environment change process.

4.4 Exit Criteria

The outputs of this procedure are:

- A completed Environment Change Request Form maintained by CM
- All information required to update the environment if the request was approved
- Information necessary to update the applicable NRC inventory

The results of the procedure are:

- The ECR is approved; a plan and schedule have been prepared and placed under CM control.
- Alternatively, the originator understands why the change request was rejected

[4] See first footnote.

4.5 Verification

Quality Assurance personnel verify that this procedure is followed and that all outputs are filed.

4.6 Roles

Table 1601–1 depicts the roles responsible for each step in the Environment Change procedure.

Table 1601–1. Environment Change Step-Role Table

Steps / Roles:	Change Request Originator	NRC Environment CCB	Change Review Committee	Change Implementation Committee	CM
Submit the ECR	P				R
Evaluate the ECR		P			S
Review the ECR	S	A	P		S
Plan and Schedule the Implementation	S	A	R	P	S
Provide ECR Status Notification		P			P

Legend: P=Performs, R=Reviews, A=Approves, S=Supports

Note: The NRC CIO (or his or her designee) appoints the members of the NRC Environment CCB. The chairman of the Environment CCB appoints the members of the Change Review Committee and the Change Implementation Committee.

Quality and Configuration Management
Series 2000

Subject	Type	Procedure
Peer Review	Identifier	P–2101
	Effective Date	October 1997
	Revision No.	

Approval C E Fitzgerald
CISSCO Program Director

1. PURPOSE

This procedure specifies how to plan for and conduct a peer review.

Peer reviews are objective approaches to finding problems in the products of a project, at a time when correcting these errors reduces the cost of the error. They are conducted by the peers of the author who prepared the product being reviewed. When all problems found are resolved, that is, all defects are corrected and all action items are addressed, product quality is certified.

2. APPLICABILITY

This procedure applies to all NRC projects subject to the SDLCM Methodology.

Peer reviews cover potentially all technical and management products on a project. For example, peer reviews apply to the products of systems development, software development, database development, user interface development, hardware development or integration, and testing. They apply as well to planning documents, requirement documents, design documents, and test documents. Any product may be considered for peer review.

3. REFERENCE PUBLICATIONS

- *SDLCM Methodology Handbook*

4. PROCEDURE

4.1 Process Flow Diagram

The peer review procedure consists of those steps identified in Figure 2101–1.

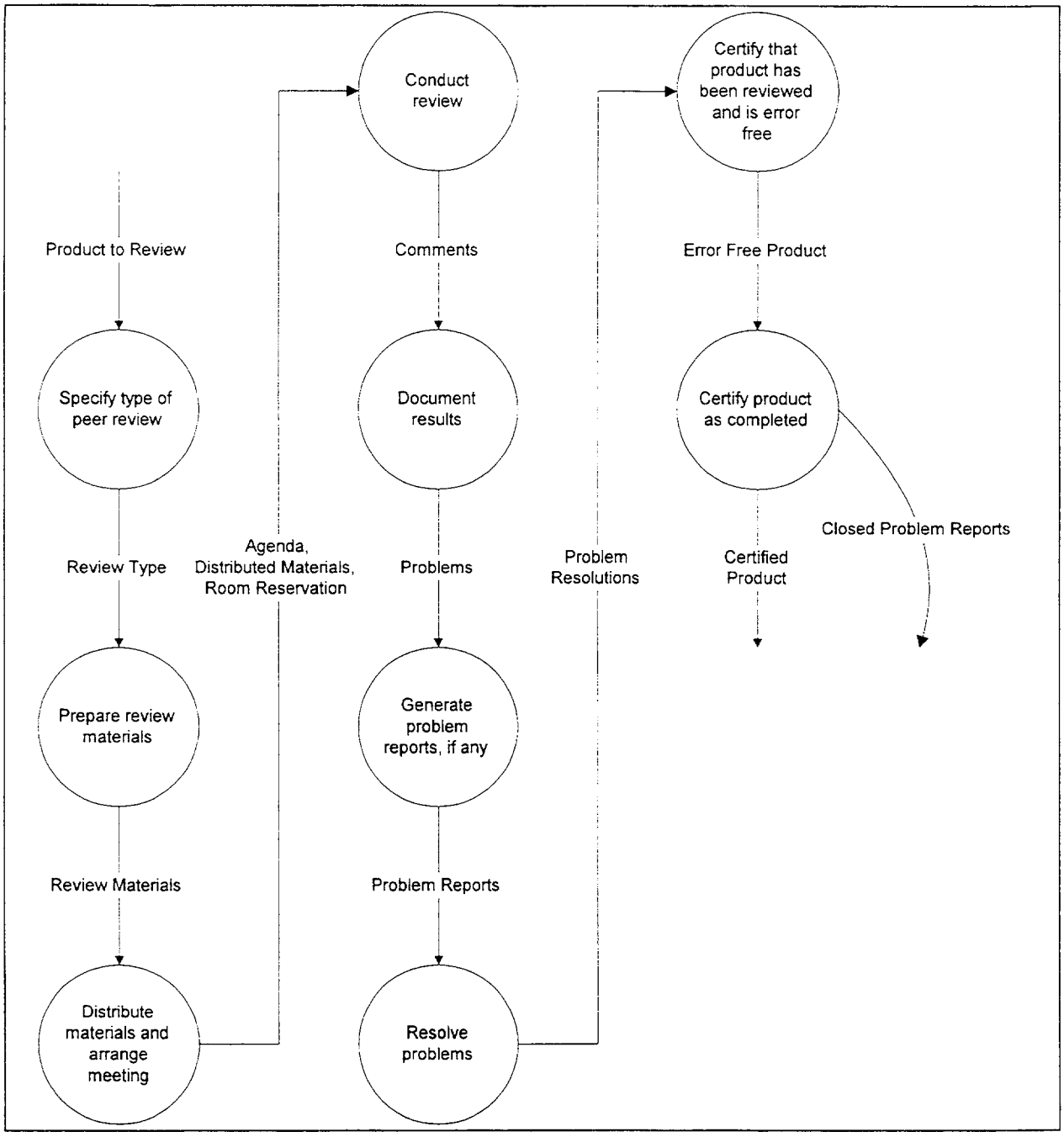

Figure 2101–1. Peer Review Procedure

4.2 Entry Criteria

The following inputs are necessary to begin this procedure:

- The product to be reviewed

The following triggers are necessary to begin this procedure:

- The product is considered to be complete by the author

4.3 Steps

Peer reviews typically involve several people—at least one product author, one moderator, and one or (usually) more reviewers.

The product author generates the material to be reviewed and makes it available to the moderator and reviewers before the inspection meeting. The moderator possesses skills and training in the dynamics of conducting reviews. The moderator has several responsibilities including keeping control of the inspection meeting. Reviewers are peers of the person whose product is being reviewed. QA may be part of the reviewers. Reviewers identify potential problems during their independent review before the inspection meeting. During the meeting the potential problems are discussed and recorded if the reviews concur. The inspection peer review is highly structured. The main focus is the identification of problems not the correction of problems. Problem resolution is the responsibility of the author. The moderator ensures that a record is kept of the resolution of all problems; the moderator may actually do this recording or the moderator may assign another, and sometimes independent, person to keep the record.

Inspection peer reviews can be used for any kind of technical or management product on a project. When all problems found are resolved, a peer certifies product quality. Until all problems found are corrected, problem reports, action item lists and other methods are used to keep the current status of the product.

1. Specify the type of peer review: one-on-one or group.
2. Prepare the review materials. The author of the product prepares the review materials, including copies of the products and any review questions or criteria that the author feels should be examined closely.
3. Distribute materials and arrange meeting. The author of the product distributes the review materials, coordinates the review schedule with participants, and reserves the meeting room. Reviewers review the product before the meeting so that the meeting time can be used efficiently.
4. Conduct the review meeting. The moderator leads the meeting and solicits comments form the reviewers.
5. Document the review results in meeting minutes.
6. Document problems in problem reports, if found.
7. Resolve problems: the author of the product fixes any problems found in the product.
8. Certify that the product has been reviewed after fixes have been made and that the product is free of errors. After all errors (if any) have been fixed by the author, a peer of the author certifies that the product is free of errors.
9. Certify that product is complete and of high quality. After the author's peer certifies that all errors have been fixed, QA certifies that the product is complete.

4.4 Exit Criteria

The results of the peer review are:

- All product problems have been resolved.

The outputs of the peer review are:

- Certified product
- Closed problem reports

4.5 Verification

Quality assurance (QA) verifies that the unit test has been performed and all problems resolved.

4.6 Roles

Table 2101–1 specifies the roles and responsibilities for each of the steps in the peer review procedure.

Table 2101–1. Unit Test Step-Role Table

Steps	Project Manager	Moder-ator	Author	Review-ers (Peers)	QA
1. Identify the product to be reviewed.	P				
2. Specify the type of peer review to be used, based on the criticality of the product.	P				
3. Prepare the review materials.			P		
4. Distribute the materials and arrange the meeting..			P		
5. Conduct the review.		P	P	P	
6. Document the review results in meeting minutes.			P		
7. Document problems in problem reports, if any.			P		
8. Resolve problems.			P		
9. Certify that product has been reviewed and all errors have been fixed.				P	
10. Certify product as complete.					P

Legend: P=Performs, R=Reviews, A=Approves, S=Supports

Subject	Type	Procedure
Structured Walkthrough	Identifier	P–2102
	Effective Date	October 1997
	Revision No.	

Approval C E Fitzgerald
CISSCO Program Director

1. PURPOSE

This procedure provides direction on how to hold a structured walkthrough to review products. Depending on whether the purpose of the walkthrough is to certify the products or to educate the staff, the rework of the material may or may not be part of the walkthrough.

2. APPLICABILITY

This procedure applies to all NRC projects, subject to the SDLCM methodology, that use peer review methods to review products.

This procedure applies to when

- The peer review type chosen for a product is a structured walkthrough.
- The walkthrough is used to validate the logical design of a system being developed or enhanced.
- A walkthrough is used to educate and get feedback.

3. REFERENCE PUBLICATIONS

- *SDLCM Methodology Handbook*
- SDLCM Methodology Procedure P–2101, Peer Review
- SDLCM Methodology Procedure F–2251, Problem Report Form

4. PROCEDURE

4.1 Data Flow Diagram

The structured walkthrough procedure comprises the three major steps identified in the data flow diagram shown in Figure 2102–1.

4.2 Entry Criteria

The structured walkthrough process shown in Figure 2102–1 has three possible triggers:

- Product, portion of product, or set of products are completed
- Logical design is completed

- New concept or technology is ready for review or demonstration and feedback

The input to the structured walkthrough process is the product to be previewed.

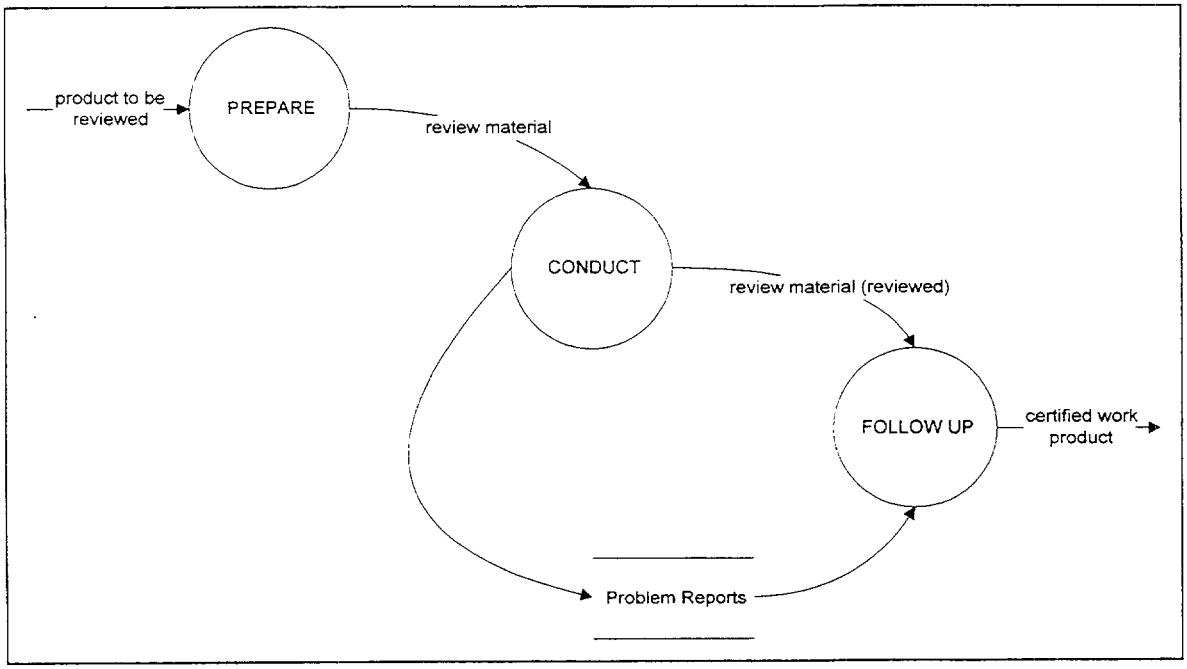

Figure 2102–1. Structured Walkthrough Data Flow Diagram

4.3 Steps

Perform the following three steps:

1. Prepare for the Walkthrough
   - Determine who the reviewers will be, and assign a scribe from among the reviewers.
   - Assemble the review material. When changed material is being reviewed, give the rationale for the change (such as by providing a problem report number). When tool support is available, also mark exactly where the changes have been made.
   - Schedule the review. Notify the reviewers of the date, time, and place.
2. Conduct the Walkthrough
   - Distribute the review material. If the product is to be reviewed with a demonstration, there may not be any material to distribute.
   - Walk through the material, discussing each product in the review package. Answer questions regarding the material.
   - Identify errors and issues.

- Fill out a Problem Report Form (F–2251) to record all errors identified and issues to be resolved. Include all errors discovered in the problem report, whether or not they are within the scope of the review.

3. Follow-up on the Walkthrough
   - Assign responsibility for resolution of errors and issues
     - If any errors found are outside of the scope of the review (for example, a requirements issue during a code review or an error in a product not being reviewed), record the error and bring it to the attention of the responsible group or person.
     - If the walkthrough is not being held to certify material, assign responsibility for resolution of each problem identified to an individual or group.
     - If the walkthrough is being held in order to certify material, assign responsibility for resolution of each problem or error to the product's author(s)and proceed with the following substeps.
   - Rework the material to address recorded errors.
   - Assign someone to review and certify the product.
   - Review the rework. Continue to rework the product until all errors have been fixed.
   - Record the certification, storing it in the project's library.

4.4 Exit Criteria

The results of the review are

- The certification is placed into the project's library (only when the review is used for certification)
- Errors and issues that were identified but are outside the scope of the review are recorded in the project's library.

The output of the review (when the review is used for certification) is the certified work product.

4.5 Verification

Quality Assurance (QA) personnel attend structured walkthroughs. QA checks the material to be walked through for compliance with project standards and verifies that the walkthrough process is being followed. If the review is used for certification, QA checks the certification audit trail.

4.6 Roles

Table 2102–1 depicts the roles responsible for each step of the Structured Walkthrough procedure.

Table 2102–1. Structured Walkthrough Step-Role Table

Roles: / Steps:	Team Lead	Author	Reviewer	Scribe	QA	Follow-Up Reviewer
1. Prepare for Walkthrough						
Assign reviewers and scribe	P					
Assemble materials	R, A	P				
Schedule the review and notify reviewers	P					
2. Conduct the Walkthrough						
Distribute materials		P				
Walkthrough materials and answer questions		P				
Identify errors and issues			P		P	
Record errors and issues	R, A	R	R	P	R	
3. Follow-Up on the Walkthrough						
Assign responsibility for resolution of errors and issues	P					
Rework the material to address the recorded errors		P			R	R
Assign reviewer	P					
Review and certify product						P
Record Certification	P					

Legend: P=Performs, R=Reviews, A=Approves, S=Supports

Subject	Type	Procedure
Operational Readiness Review	Identifier	P–2141
	Effective Date	October 1997
	Revision No.	

Approval C E Fitzgerald
CISSCO Program Director

1. PURPOSE

This procedure specifies how to plan for and conduct an operational readiness review (ORR).

An ORR is a formal life-cycle review. Its purpose is to determine whether a configuration item (CI), or a collection of CIs composing a system, is ready to be released for operations and maintenance (O&M). Those attending include project managers, the engineering team (representatives from systems, hardware, and software engineering); acceptance test and maintenance teams; users; and the client.

The ORR occurs after completion of acceptance and operational testing but before the beginning of full-scale operations. Allow enough time between the ORR and start of operations to permit the resolution of problems that might occur during the phase in of full-scale operations. If possible, conduct the ORR at least 90 days before the start of operations.

2. APPLICABILITY

This procedure applies to all NRC projects subject to the SDLCM Methodology.

This procedure applies to all projects that prepare systems for operational support after acceptance testing.

3. REFERENCE PUBLICATIONS

- *SDLCM Methodology Handbook*

4. PROCEDURE

4.1 Process Flow Diagram

The ORR procedure consists of those steps identified in Figure 2141–1.

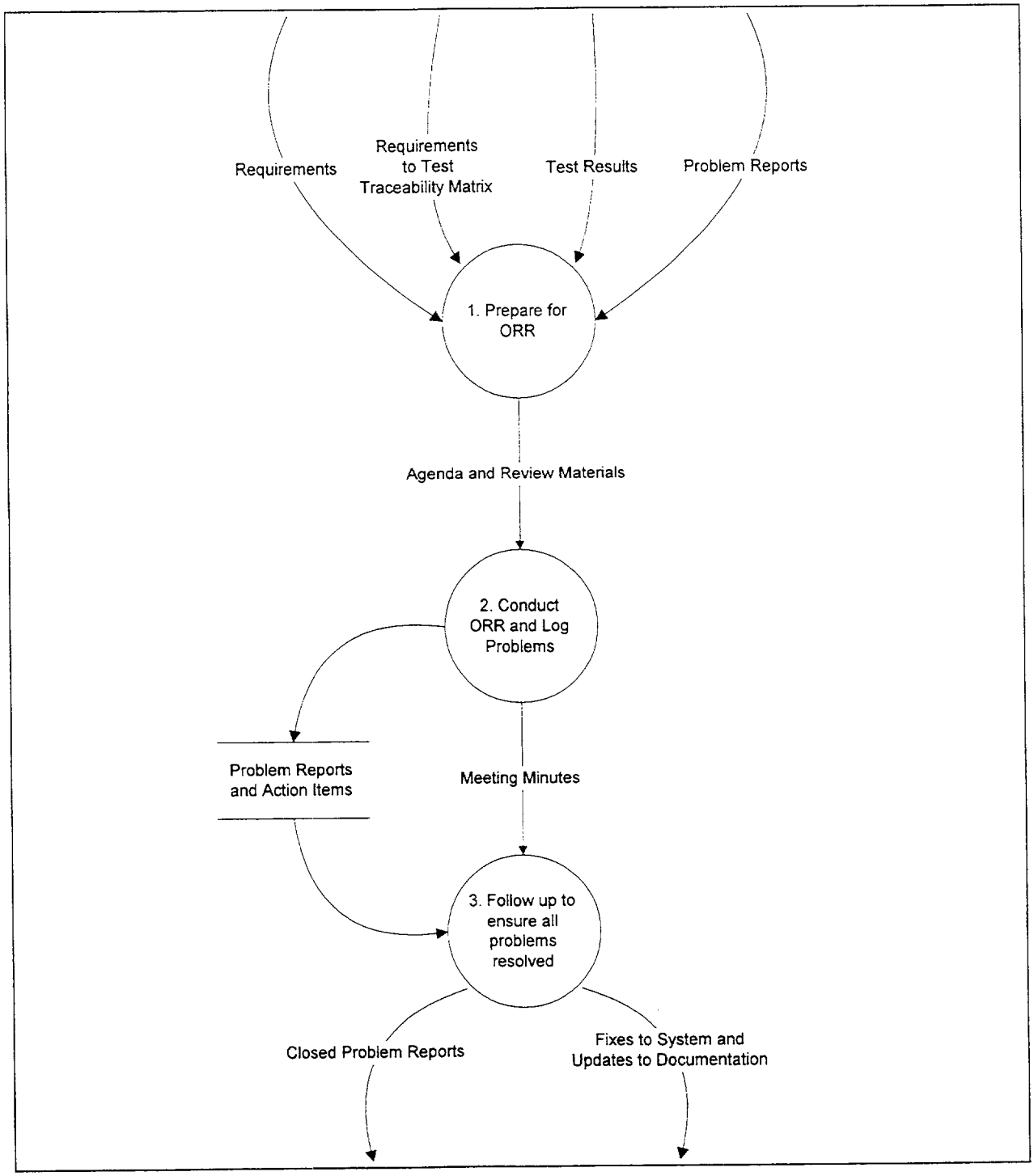

Figure 2141–1. ORR Procedure

4.2 Entry Criteria

The following inputs are necessary to begin this procedure:

- Requirements (Project Definition and Analysis Document)
- Requirements to Test Traceability Matrix
- Test Results
- Problem Reports

The following triggers are necessary to begin this procedure:

- The system has passed acceptance and operational testing.

4.3 Steps

1. Prepare for ORR. Gather and prepare materials for the ORR, and schedule the review with the participants.
2. Present information and log action items and problem reports for the following:
   - System Overview. Provide a high-level overview of the system from the user's viewpoint. Show interfaces between system components and with external entities. List the products generated by the system and the schedule for their delivery. Indicate key aspects of system design, such as interactive versus batch capabilities.
   - Requirements Summary. Provide a high-level summary of the system requirements. Include a discussion of the operational environment. If the system being reviewed is a new release of an existing system, emphasize new requirements.
   - Acceptance and Operational Testing. Summarize the results of acceptance and operational testing. Show the partitioning of test cases among system components and the final results for each component. Include a matrix of test cases versus system requirements. Discuss the disposition of test items that did not pass unconditionally. Refer to the document(s) that contains test plans and results.
   - Documentation Status. Give the status of all documentation that will support system O&M. Include such documents as the system and operations concept, requirements specifications, interface control documents, system description, users guide, and operations and maintenance procedures.
   - Status of Operations and Maintenance Procedures. Present the procedures for operational use and maintenance of the system, including responsibilities and approvals. If the system shares resources with other systems, assess usage of those resources (for example, central processing unit loading, disk space allocations, number of terminals). Describe maintenance items that are in progress or are required to support operations. Discuss enhancements that are

in progress or are recommended for future implementation. Provide an overview of configuration management procedures with reference to detailed procedures.

- Training Materials and Plans. Summarize progress on training operations personnel by function and name. Describe the process that will be used to train new personnel, which may include system documentation review, operations procedures demonstration, hands-on practice sessions, classroom lectures, and operations simulations. Include a training schedule. Where appropriate, define the process that will be employed to certify personnel to operate the system.
- Schedule for the Phase In of Operations. List the events and planned dates for activities that must be completed before the start of normal system operation. Include any activities that must be completed. Compare current status with the schedule presented.
- Contingency Plans. List contingencies and assess them in terms of their effects on operations (for example, operate normally, operate with degraded performance, or inability to operate). Discuss the user response to each contingency.
- Issues, Items To Be Resolved, and Problems. Discuss each issue, item to be resolved, or outstanding problem that may hinder normal system operation. Give the history of the item (for example, enhancement identified during acceptance testing), current status (for example, maintenance request prepared), criticality to operation, and expected resolution date.

3. Follow up. Ensure that all problems have been resolved prior to designating that the ORR is completed.

4.4 Exit Criteria

The outputs of the ORR are:

- Resolved problems (closed problem reports)
- Any fixes to the system or documentation as required by the ORR.

The results of the ORR are:

- All problems have been resolved.
- The system is ready to deploy.

4.5 Verification

Quality assurance (QA) verifies that the ORR has been performed and all problems resolved.

4.6 Roles

Table 2141–1 specifies the roles and responsibilities for each of the steps in the ORR procedure.

Table 2141–1. Unit Test Step-Role Table

Steps	Project Manager	Engineering Team	Test Team	O&M Team	Users	Customer
Prepare for the ORR.	P	P	P	P		
Present information and log action items and problem reports.						
• Present the system.		P			R	A
• Present the requirements.		P			R	A
• Present the results of acceptance and operational testing.			P		R	A
• Present the documentation status.		P			R	A
• Present the status of O&M procedures.		P			R	A
• Present the status of training materials and plans.		P			R	A
• Discuss the schedule for the phase in of the system for operations.	P	P			R	A
• Discuss contingency plans if the system does not perform as expected.		P			R	A
• Discuss issues, items to be resolved, and problems.		P	P	P	R	A
Follow up by reviewing action item status and problem reports. Decide if the system can be deployed.	A	P	P	P		A

Legend: P=Performs, R=Reviews, A=Approves, S=Supports

Subject	Type	Procedure
Configuration Control Board	Identifier	P–2501
	Effective Date	October 1997
	Revision No.	

Approval C E Fitzgerald
CISSCO Program Director

1. PURPOSE

The purpose of this procedure is to document the review process used by a Configuration Control Board (CCB).

2. APPLICABILITY

A Configuration Control Board provides a structured review process of requirements changes, system changes, problem areas, and work in progress. Any member of the CISSCO program may submit a change request via the established management chain.

3. REFERENCE PUBLICATIONS

The following publication contains related information:

- *SDLCM Methodology Handbook*
- SDLCM Methodology Form F–2251, Problem Report
- SDLCM Methodology Procedure P–2502, Change Proposal
- SDLCM Methodology Form F–2502, Change Proposal Form
- SDLCM Methodology Procedure P–9001, SDLCM Methodology Change
- SDLCM Methodology Form F–9001, SDLCM Methodology Change Request Form

4. PROCEDURE

4.1 Data Flow Diagram

The CCB procedure includes the 5 steps identified in the data flow diagram shown in Figure 2501–1.

4.2 Entry Criteria

A properly completed change instrument, for example, a Change Proposal (F–2502) or Problem Report (F–2251), is submitted to CM and placed on the CCB agenda.

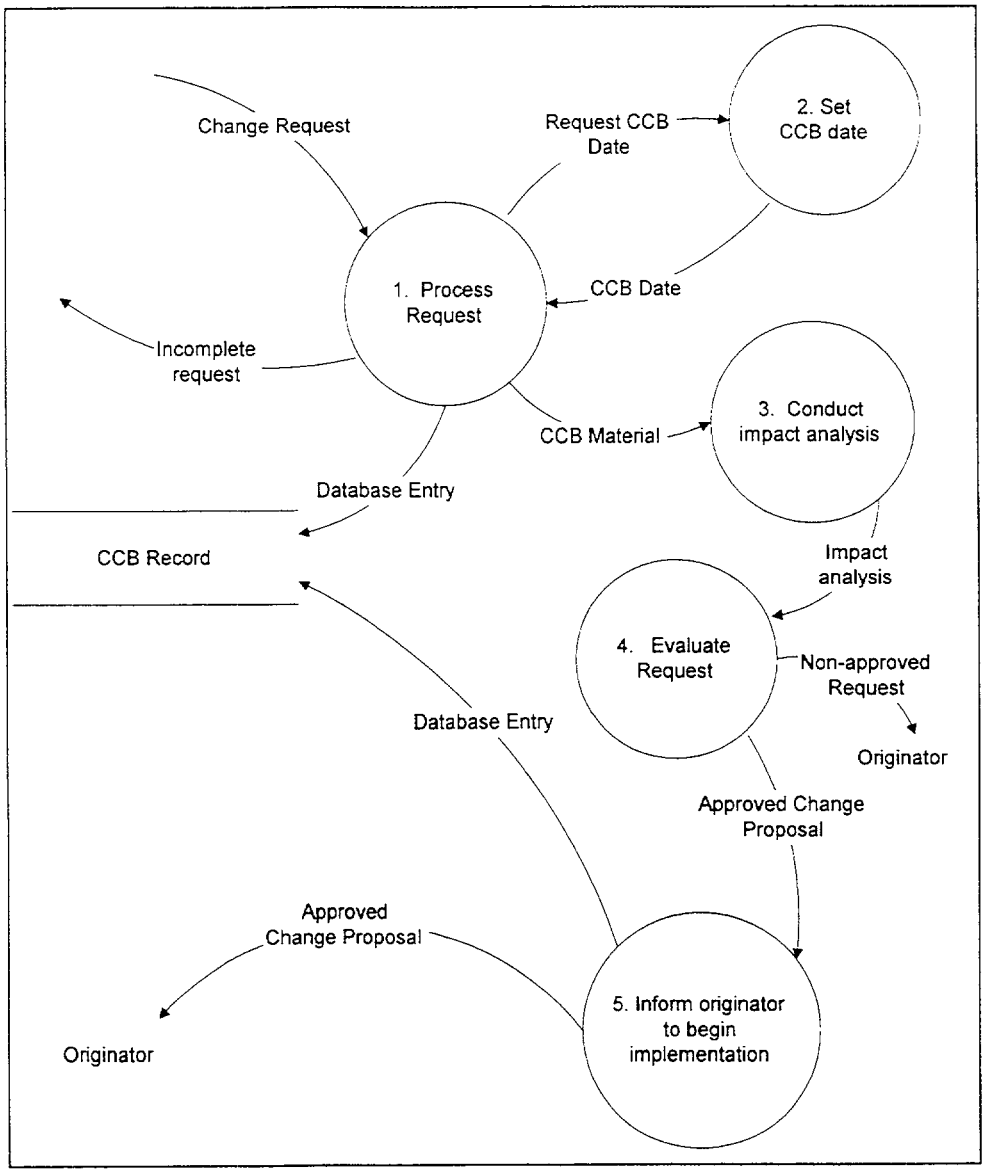

Figure 2501–1. CCB Data Flow

4.3 Steps

The CCB evaluates all requested changes. All CCB decisions must be consistent with the project policies, priorities, and schedules. The CCB ensures proper and timely consideration of all requested changes. The following steps define the CCB process.

1. Configuration Management (CM) receives all requests for change such as Change Proposals (CPs) and Problem Reports. CM reviews the requests for completeness and clarity. If the requests are not satisfactory, CM identifies missing data and returns the requests to the appropriate technical manager for completion. CM marks the requests

with control numbers according to the procedures for individual request types and logs them into the appropriate database.

2. CM coordinates and schedules a date to hold the CCB with the appropriate manager and assembles a CCB meeting package. The package consists of the following items:
   - CCB notification memo with scheduled time and date of CCB
   - Agenda of CCB activities
   - Change request proposals
   - The notification memo and agenda are normally distributed to participants three working days prior to the meeting
3. The CCB members review the material and ensure an impact analysis is conducted by individuals who have the required technical and managerial expertise to assess the impact of the change. The impact analysis is based on the considerations identified in Table 2501–1.
4. If impact is identified:
   - The CCB reviews the impact assessment
   - The CCB members recommend approval, disapproval, or deferment of each item to the Chairperson
     - Approved requests are designated for action,
     - Disapproved requests are designated for archival
     - Deferred requests are designated for further analysis
   - The CCB Chairperson makes final determination
   - CM records CCB decisions in the CCB control record
5. CM prepares official CCB meeting minutes and notifies the request originator of the CCB decision.

4.4 Exit Criteria

When all requests have been evaluated and decisions recorded, the chair adjourns the meeting.

4.5 Verification

The Quality Assurance organization audits the CM libraries and directories to ensure all documentation and software products required by this and any other applicable procedure are being followed.

4.6 Roles

Table 2501–2 specifies the roles and responsibilities for each of the steps in the CCB procedure.

Table 2501–1. Impact Analysis Checklist

1. Is the change necessary?
2. What is the benefit of the change?
3. What is the effect of the change on performance and system capacity?
4. What is the priority of the change?
5. What is the relationship of the change in the effected CSCIs to other CSCIs or CSCs?
6. How will the change be implemented?
7. Who will implement the change?
8. When will the change be implemented?
9. Where will the change be implemented?
10. How long will it take to implement the change?
11. How will the change affect other scheduled activities?
12. What is the cost of the change in terms of material and labor force?
13. What is the impact on technical documentation such as diagrams, specifications, design documentation, and user manuals?
14. What is the impact on training programs?
15. Is the change cost effective?
16. What is the impact on user interface and operations?

Table 2501–2. Configuration Control Board Step-Role Table

Roles: / Steps:	CCB Chair	CCB Members	Responsible Manager	CM	QA
Submit Change Proposal (CP)			P	R	
Schedule CCB, add CP to agenda	A			P	
Conduct Impact analysis		P	S	S	
Evaluate request	A	P	S	S	R
Implement Change	A		P	S	S

Legend: P=Performs, R=Reviews, A=Approves, S=Supports

Subject	Type	Procedure
Change Proposal	Identifier	P–2502
	Effective Date	October 1997
	Revision No.	

Approval C E Fitzgerald
CISSCO Program Director

1. PURPOSE

This procedure establishes the mechanism for proposing changes to baselined application or information systems. It describes the process for submitting a Change Proposal (CP) Form. Any change to baseline software or documentation that affects cost, schedule, or technical requirements must be authorized by the Nuclear Regulatory Commission (NRC) prior to implementation.

2. APPLICABILITY

This procedure applies to all proposals to change baselined application or information systems software or documentation.

3. REFERENCE PUBLICATIONS

The following publications contain related information:

- *SDLCM Methodology Handbook*
- SDLCM Methodology Procedure P–2501, Configuration Control Board
- SDLCM Methodology Form F–2502, Change Proposal Form

4. PROCEDURE

4.1 Data Flow Diagram

The Change Proposal procedure has the five major steps identified in the data flow diagram shown in Figure 2502–1.

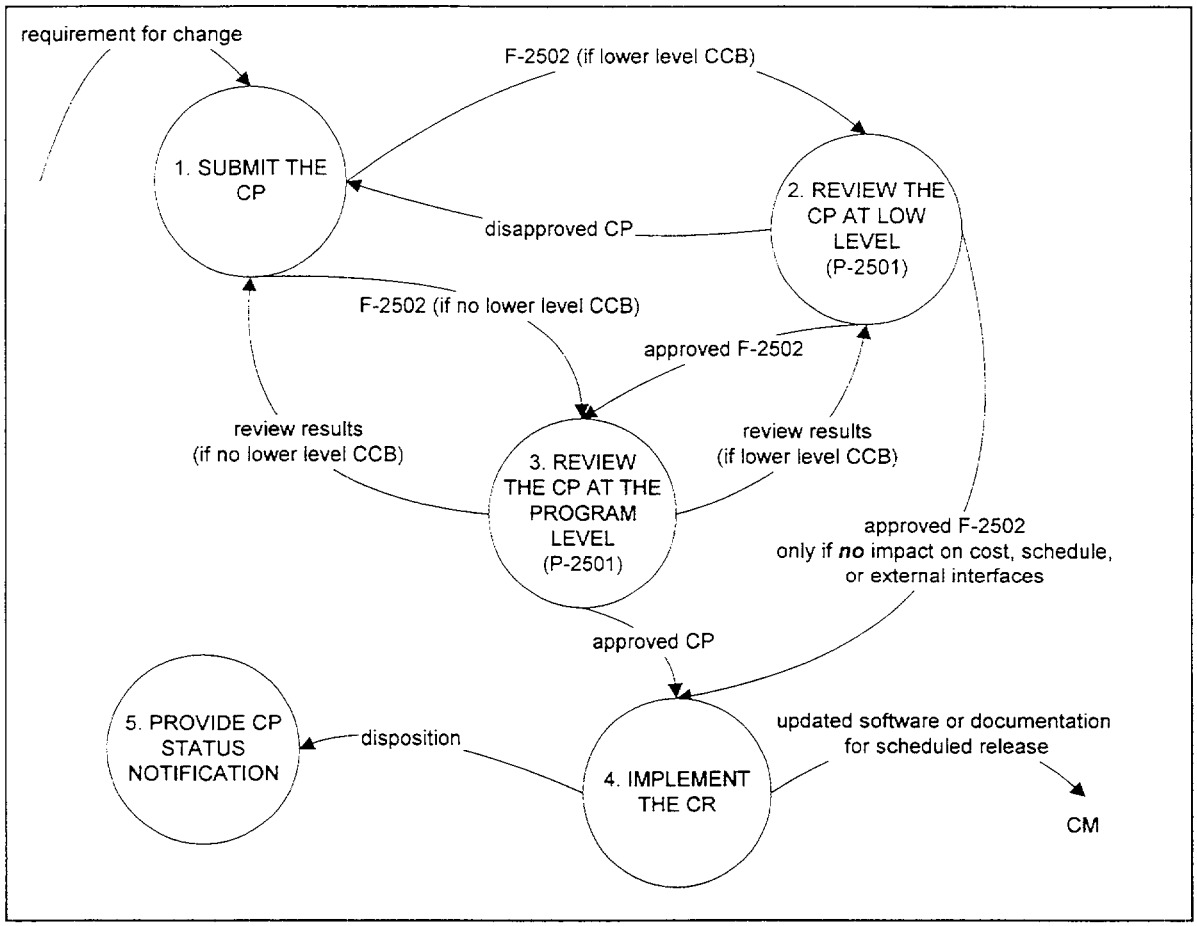

Figure 2502–1. Change Proposal Data Flow Diagram

4.2 Entry Criteria

The following input is necessary to begin this procedure:

- A recognized need for an addition, deletion, or modification to an existing baselined application or information system or its documentation set

Either of the following events may trigger this procedure:

- A new system requirement
- The introduction of new technology

4.3 Steps

Perform the following steps:

1. Submit the Change Proposal

   Any NRC personnel, member of the user community, or personnel from NRC contractor organizations may submit a CP.

To propose a change, use Change Proposal Form F–2502. Complete all blocks under "Proposal Originator Information" and "Change Information." Forward the CP Form to the originator's immediate supervisor for approval. The supervisor submits the form to the CM organization for processing as defined in SDLCM Methodology Procedure P–2501, Configuration Control Board.

If a CCB exists at the system, project, or Task Area level, CM submits the CP Form to the lower level CCB; proceed to Step 2. If a lower level CCB does not exist for this application or information system, CM submits the proposal directly to the CISSCO Program level CCB for review; proceed to Step 3.

2. Review the proposal at the low level (if applicable)

   The system, project, or Task Area level CCB reviews the proposal (see Procedure P–2501).

   If the proposal is disapproved, the CCB returns the CP Form to the originator. If it is approved, this CCB forwards the form to the CISSCO Program level CCB for review (Step 3). Note that the lower level CCB may forward the approved CP directly to Step 4 for implementation if and only if (a) the change does *not* affect cost, schedule, or external interfaces and (b) the lower level CCB has an appropriate agreement in place with the CISSCO Program CCB to permit bypassing Step 3.

3. Review the proposal at the CISSCO Program level

   The Program level CCB reviews the CP (see Procedure P–2501).

   If the CCB approves the proposal without change, it is passed to Step 4 for implementation. If the CCB (a) disapproves the proposal, (b) defers implementation of the proposal to a later date, or (c) approves the proposal with changes, then the CP is returned either to the lower level CCB (if one exists) or to the originator.

4. Implement the approved change

   Follow CISSCO Program procedures to issue a statement of work and provide funding to implement the approved change using a suitable contract vehicle. The process for maintaining or enhancing an existing application or information system is described in the *SDLCM Methodology Handbook*. If the approved change is for a current project developing a new system, follow program level procedures to issue an appropriate modification.

   Deliver the final product to the Configuration Management (CM) organization for inclusion in the next release.

5. Provide CP status notification

   CM notifies the originator of the final disposition of the CP.

4.4 Exit Criteria

The outputs of this procedure are:

- A completed Change Proposal Form

- Materials required to update the system software and documentation at the scheduled release

The results of the procedure are:

- The application or information system supports the implemented CP
- Alternatively, the originator understands why the CP was rejected

4.5 Verification

Quality Assurance audits the CM libraries and directories to ensure that all steps have been followed and that all documentation and products required by this procedure are produced.

4.6 Roles

Table 2502–1 depicts the roles responsible for each step in the Change Proposal procedure.

Table 2502–1. Change Proposal Step-Role Table

Roles: / Steps	Change Proposal Originator	System, Project, or Task Area CCB	CISSCO Program Level CCB	CISSCO Contractor	CM
Submit the CP	P			S	R
Review the CP at low level		P			S
Review the CP at program level			P		S
Implement the CP				P	S
Provide CP Status Notification					P

Legend: P=Performs, R=Reviews, A=Approves, S=Supports

Requirements and Design
Series 3000

Subject	Type	Procedure
Software Engineering Notebook Processing	Identifier	P–3041
	Effective Date	November 1998
	Revision No.	1

Approval C E Fitzgerald
CISSCO Program Director

1. PURPOSE

This procedure documents the processing of a Software Engineering Notebook (SEN). An SEN organizes and centralizes information pertaining to an element of software and ensures ready access to that information for modification and audit. See SDLCM Methodology Standard S–3091, Software Engineering Notebook, for more information on content.

2. APPLICABILITY

This procedure applies to all software elements (for example, modules, units, data files, database subschema), whether new, adapted, or converted, composing a software system under development or in maintenance. This procedure does not necessarily apply to unchanged transported software elements.

3. REFERENCE PUBLICATIONS

The following publications contain related information:

- *SDLCM Methodology Handbook*
- SDLCM Methodology Standard S–3091, Software Engineering Notebook

4. PROCEDURE

SENs are created when the Software Requirements Specification (SRS) is baselined and maintained through the final delivery of the software

4.1 Data Flow Diagram

An illustration of the SEN process is shown in Figure 3041–1.

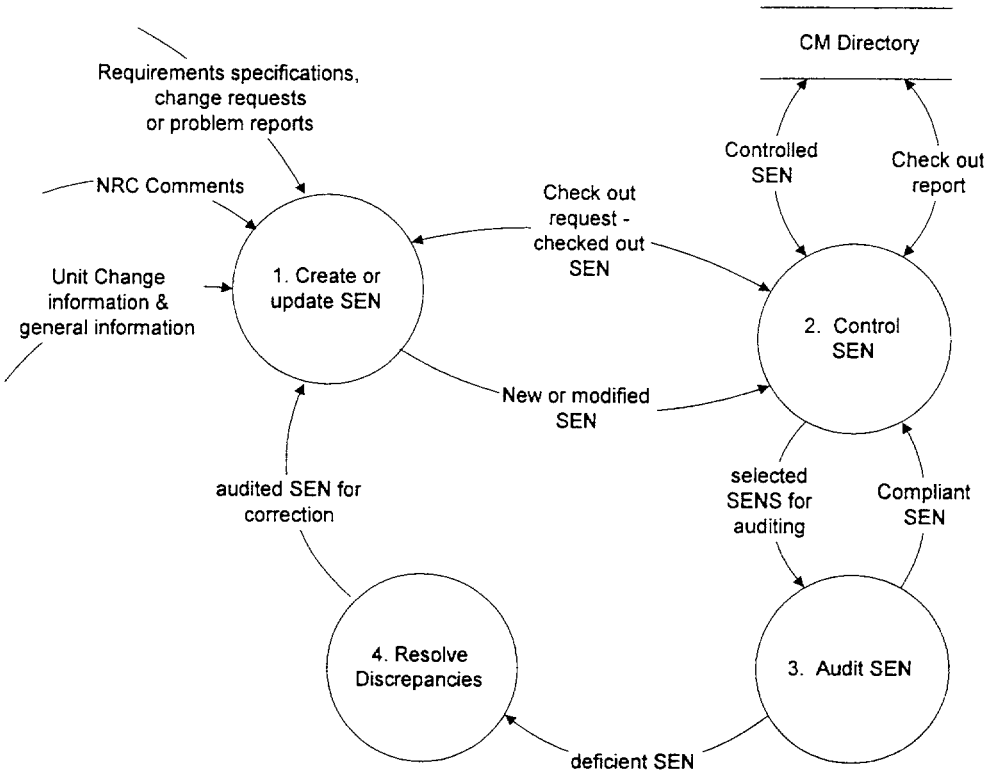

Figure 3041–1. Software Engineering Notebook Processing Data Flow Diagram

4.2 Entry Criteria

An SEN will be created for each new module during the design phase and will be maintained through completion. While a module is being designed, coded, and tested, the SEN will be used as a workbook and maintained by the programmer working on that module or system. An SEN will be created and maintained for each legacy software element whenever it becomes necessary to modify the element.

4.3 Steps

1. Create or update the SEN:
    - For new modules, initiate the SEN with the original software specifications and design data and all pertinent information through deployment. (See the sample checklist in Table 3041–1. See SDLCM Methodology Standard S–3091 for information about the content of the SEN.)
    - For modifications to existing software, check the current SEN out of the CM library and maintain it in a current status reflecting all new modifications.

- For modifications to transported software with no SEN, create an SEN using existing documentation and other available information that coincides with the SEN checklist.
- When the module has been deployed, deliver the SEN to CM for maintenance.

2. CM maintains the SENs in a controlled library. CM maintains a record of checked out and returned SENs.
3. QA may audit an SEN at any time during the software development life cycle to verify that its contents are complete for that particular phase.
4. SEN audit results are provided to the appropriate software development manager (current or last user) to resolve any discrepancies.

Table 3041–1. Software Engineering Notebook Processing Checklist

Checklist Item	Comments	YES	NO	N/A
Prologue • System/program description (including version) • Inputs • Outputs • Interfaces • Changes (TAC # or Problem Report, title, date of change)				
Design documentation: • Detailed design • Inspection Checklists • PDL				
Current listing (s)				
Test plans and procedures				
Test results				
Problem reports (corrected)				
Difference listing(s) (previous vs. current)				
Build procedures				
Installation instructions				

4.4 Exit Criteria

After the software development life cycle is complete (software delivered to NRC for acceptance testing), the developer submits the SEN to CM and the SEN is added to the controlled CM library.

4.5 Verification

QA performs a random sample audit as part of the normal CM audit or a build release audit.

4.6 Roles

Table 3041–2 is a Step-Role Table for the Software Engineering Notebook Processing Procedure.

Table 3041–2. SEN Processing Step-Role Table

Roles: / Steps:	SW Engineer	QA	CM
Create or update SEN	P		
Control and check out SEN		R	P
Audit SEN		P	S
Respond to QA Audit	P	A	

Legend: P=Performs, A=Approves, R=Reviews, S=Supports

Subject	Type	Standard
Project Definition and Analysis Document	Identifier	S–3051
	Effective Date	November 1998
	Revision No.	1

Approval C E Fitzgerald
CISSCO Program Director

A. PURPOSE

This standard specifies content and format requirements for the Project Definition and Analysis (PDA) Document.

B. APPLICABILITY

This standard applies to all projects that fall under the SDLCM Methodology umbrella. It is used by those persons who define initial project requirements to establish the scope of the project, identify the requirements, and analyze alternative solutions.

C. REFERENCE PUBLICATIONS

The following publications contain related information:

- *SDLCM Methodology Handbook*, Component 1
- SDLCM Methodology Standard S–1051, Project Charter
- SDLCM Methodology Standard S–3052, Current System Assessment Document
- SDLCM Methodology Standard S–3162, Context Diagrams
- SDLCM Methodology Standard S–3171, Logical Design Document
- Project Initiation Plan (if project is a result of Enterprise Integration/Migration activities)

D. STANDARD

Use the PDA Document to summarize the results of the project definition and analysis activities of SDLCM Methodology Component 1 into an organized and logically flowing deliverable. The following paragraphs describe the content of each section.

Note that the PDA Document is a working document that is created initially as an activity within Component 1 of the SDLCM Methodology. The PDA Document is updated at defined milestones during the project's life cycle and as necessary as the project matures.

1. INTRODUCTION

1.1 Background

Briefly describe the project relative to the client's mission. Explain why the project has been initiated, its importance to the client, the specific results or long-term objectives the client wants to achieve, and the project's contribution to the overall strategy. Also describe the problem to be solved or the product to be developed in enough detail to support planning.

1.2 Objectives

Specify the client's business objectives that this project is to support, the critical functions the project should achieve, and the quantifiable criteria the project must meet to succeed.

1.3 Scope

Specify those aspects of the client's situation—such as customers, products, processes, organizations, locations, or applications—that are to be included in the project and those that are to be excluded. For example, if the project is to develop a new system, this section describes the scope of the proposed system in terms of the business processes or functions to be automated.

Describe the external influences and impacts—such as interfaces, customer needs, and regulatory requirements—that are to be addressed by the project.

Specify any constraints, or restrictions, associated with the project. A constraint may relate to project approach, priorities, personnel, time, technologies, environments, decision cycles, tools and techniques, or other aspects of the project. Discuss how management will control identified constraints to ensure project success.

1.4 Assumptions

Specify the major assumptions used to establish the project estimates, plans, and approach. Assumptions are expectations, which have not been stated explicitly, that form the basis for project decisions. Assumptions are also necessary for each identified issue if the project is to continue while issues remain unresolved. In this case, include the degree of criticality with the description of each assumption.

1.5 Applicable Documents

List any other documents that apply.

1.6 Overview

Table 3051–1 shows the evolution of the PDA Document, using the Project Charter as a beginning point.

Table 3051–1. PDA Document's Component Sections in Relationship to Component 1 Activities

ACTIVITIES	DELIVERABLES
Clarify Scope of the Project	Project Definition and Analysis Document • Background • Objectives • Scope
Identify Functional & Data Requirements	• System Requirements • Data Requirements ⇒ Context Diagram ⇒ Entity List ⇒ Entity Definitions
Analyze Alternative Solutions	• Current System Assessment Document • Analysis of Alternatives • System Operations Concept

2. APPROACH

Describe the general approach, delivery methodology, project management approach, and standards to be employed on the project. Identify any exceptions, and approvals of any exceptions, to policies, procedures, and standards that apply to this project.

3. SYSTEM REQUIREMENTS SPECIFICATION (SRS)

The purpose of this section is to define the system requirements specification (SRS). An SRS specifies the functional, performance, operational, and programmatic requirements for a system as an entity.

State individual requirements in quantitative terms. Avoid the use of negative requirements. Consider testability, completeness, consistency, and conciseness when writing each requirement.

Number all requirements uniquely to facilitate traceability.

3.1 Functional Requirements

Identify all system functional requirements based on the system and operations concept and other user needs or client requirements.

Organize the requirements into functional areas. For each functional area, define, as appropriate:

- Input data
- Process(es) (algorithms, transformations, manipulations, calculations)
- Output data

- External interfaces
- Communications
- Special management information needs

Use data requirements and system control as section headings, if appropriate.

Include a separate subsection for each functional area.

3.2 Performance Requirements

Decompose the performance requirements section into the following subsections, as appropriate:

- External workloads (for example, total size of data input per day)
- Internal function workloads (for example, function X must compute responses within 18 milliseconds)
- Throughput and response times (for example, complete simple query in the designated time period)
- Data quality, integrity, accuracy (for example, some elements must be accurate to a certain number of significant figures)
- Data retention (for example, number of years to retain data on-line versus long-term storage)
- System and communications capacity (for example, maximum number of communications lines connected to the system)
- Reliability, maintainability, and availability (for example, mean time between failures, mean time to repair)
- Periodicity, precision, simultaneity, frequency of transactions, processing, operations (for example, the system must process 10,000 transactions per second)
- Human workload and performance (for example, generation of a designated number of reports per day)
- Growth, flexibility, and expandability (for example, the system shall be capable of supporting four additional interfaces)
- Fault detection and isolation

3.3 Operational Requirements

Decompose the operational requirements section into the following subsections, as appropriate:

- Human factors (for example, user interface, keyboard layouts, display panel design, display formats).
- Environment (for example, temperature, humidity, air conditioning, shock and motion, shelf life)
- System monitoring
- Configuration control (for example, automated configuration control software)

- Training (for example, operator training)
- Support capabilities (for example, checkpoint and restart)
- Maintenance (for example, preventive maintenance)
- Logistics (for example, tracking of magnetic tapes)
- Facilities (for example, government-furnished equipment, test equipment, space, environmental such as heating and air conditioning)
- Safety (for example, fire, grounding of electrical equipment, requirements intended to prevent personnel injury)
- Security (for example, facility access, password protection, software, and data security requirements)
- Geographic location (for example, multiple sites)
- Documentation (for example, users or operations guides)

3.4 Programmatic Requirements

Decompose the programmatic requirements section into the following subsections, as appropriate:

- Development facility
- Development support requirements or constraints
- Special test requirements
- Installation, phase-in, and turnover
- Schedule and budget constraints
- Growth, flexibility, and expandability
- Development standards
- Procurement activities
- Shipping and storage requirements

3.5 Special Requirements

Specify any system requirements that are not appropriate for inclusion in any of the previous sections.

3.5.1 ACCESSIBILITY REQUIREMENTS FOR INDIVIDUALS WITH DISABILITIES.

3.5.2 RECORDS MANAGEMENT REQUIREMENTS.

4. DATA REQUIREMENTS

Define the data requirements of the project, including

- A list of entities, that is, things that you want to keep information about

- The entity definitions, that is, attributes of the entities
- A context diagram to explain the relationships among the entities.

4.1 Entity List and Definitions

Identify those top-level entities required by the project. Include only those entities shown in the context diagram and known at the functional level. (Note that the PDA Document defines initial project requirements and analyses alternative solutions. It is not a system requirements analysis and design document. A complete list of entities will be created and documented in the Logical Design Document, S–3171, as a product within Component 3, Design the Solution.)

Provide basic information for each top-level entity as an entity definition:

- Name
- Identifier (primary key)
- List of included subtypes, if any
- Description (several sentences)
- Average volume at conversion
- Annual growth percentage
- Active life
- Estimated size in characters (preliminary)
- Source of the data for the entity (existing or to be created) or method of derivation from existing or to-be-created source(s)

Compare the entity list with the Enterprise Model to ensure consistency and to avoid conflict.

4.2 Context Diagram

Provide a pictorial view of the entities and the relationships among the entities. (See SDLCM Methodology Standard S–3162, Context Diagrams.)

For each entity in the entity-relationship diagram, provide:

- Name
- Description (one or two sentences)
- Cardinality

5. ASSESSMENT OF CURRENT SYSTEM

While keeping the functional and data requirements in mind, summarize your assessment of the current system (if any) to determine what (if any) changes (additions, deletions, replacements) are required to fulfill the functional and data requirements. This is illustrated in Table 3051–2. If the current system is complex, prepare a separate Current System Assessment Document using Standard S–3052 and reference it here.

Table 3051–2. An Assessment of the Existing and Desired Functionality or Attributes of System X

Priority	Functions or Attributes Required	Functions or Attributes that are Part of the Existing System	Additional Functions or Attributes Required
	A	A	
	B	B	
	C		C

6. ANALYSIS OF ALTERNATIVES

Given the required functional and data requirements and the assessment of the current system (if any), analyze the requirements in terms of their dollar costs, effort in person hours, technical complexity and risk, and schedule demands.

Document the results of your findings of alternative ways to achieve your functional and data requirements. Possible solutions may include:

- Maintaining the application program by fixing known bugs. For example, replace the expression "2*3+4" which equals 10 with "2*(3+4)" which equals 14.
- Enhancing the application program by adding new code to the original system. For example, add a sorting function to the system.
- Complementing the current application program by, for example, adding a Commercial Off-The-Shelf (COTS) system to provide functionality that was missing in the original system. For example, add a personnel system to complement the data kept in a financial system.
- Upgrading the hardware platform, for example, to provide faster processing. For example, replace a 386 PC with a Pentium PC.
- Upgrading the operating system to provide more efficient file management. For example, replace Windows 3.1 with Windows 95.
- Replacing the system because it no longer fulfills the mission and it is less expense to replace than to repair or enhance.
- Adapting an existing system rather than starting from scratch.
- Using a COTS product as provided by a vendor rather than designing and developing a custom system.

6.1 Consideration of GSA Programs and FIP Resources

Describe which of the following considerations apply to this project (if any).

- GSA's mandatory-for-use programs.
- GSA's mandatory for consideration programs.
- GSA non-mandatory program.
- FIP resources available for reuse within the agency and from other agencies

- Existing FIP resources on a shared basis.
- Acquiring FIP resources by contracting
- Any others that may apply to your unique situation

6.2 Advantages and Disadvantages of Feasible Alternatives

Describe the advantages and disadvantages of each alternative solution by considering such things as:

- Cost in money
- Cost in Effort (Person Hours and Skill Level Required)
- Time in Calendar Time
- Speed of Performance
- Accuracy
- Risk to Accuracy or Reliability Due to Technical Complexity

For example, some variant of Table 3051–3 may be used to summarize the strengths and weakness of the identified solutions, followed by a more detailed description of the analysis.

Table 3051–3. Summary Table of Analysis of Alternatives for Project X

Alternatives (Examples)	Cost in U.S. Dollars	Cost in Effort (Person Hours)	Time in Calendar Time	Speed of Processing	Risk to Accuracy
Enhance the Application Program					
Fix the Application Program					
Upgrade the Hardware Platform					
Upgrade the Operating System					
Upgrade the COTS Application Program					
Add a Complementary Personnel System					
Replace the Obsolete Financial System					

6.3 Selection of the most advantageous alternative.

Identify the most advantageous solution, emphasizing your rationale with respect to the priorities of function and attributes of the project. For example, if accuracy and reliability are more important than cost and timeliness, state so.

7. SYSTEM OPERATIONS CONCEPT (SOC)

Summarize the results of your definitions and analyses in this section of the PDA document by presenting a conceptual description of the proposed system and its operation.

Provide initial cost and resource estimates based on the proposed system concept. Identify critical technological limitations and key cost drivers. Define the approach for tailoring the development effort to the specific work required. Define the procurement approach if major procurements are envisioned. Provide an operational description of the system. Specify

significant operational requirements, operational interfaces, operational scenarios, and operations personnel requirements.

7.1 System Description

Describe the system, providing its capabilities and other characteristics.

7.1.1 TOP-LEVEL SYSTEM VIEW

Present a top-level description of the system, its operational environment, and interfaces to provide an understanding of what the system is to accomplish.

7.1.2 FUNCTIONAL AND OPERATIONAL CAPABILITIES

Describe the major functional and operational capabilities required to meet the needs of users and operators. Identify which user and operator needs are supported by each capability. Use a table or matrix if appropriate for the sake of clarity. Discuss trade studies performed to establish how best to provide the system's required capabilities.

7.1.3 SYSTEM CHARACTERISTICS

Describe significant characteristics required of the system. Possible areas of discussion include processing capabilities; data structures; performance; error recovery; fault tolerance; reliability, maintainability, and availability; risk criticality; safety and security; flexibility and expandability; transportability; quality; and adaptability to various operational sites. Focus only on those characteristics that will have considerable impact on the system design.

7.1.4 REFERENCE ARCHITECTURE

Partition system functions and operational capabilities into logical groupings, and allocate system capabilities to those logical groupings. Highlight key system characteristics. Identify existing system components and any required modifications.

7.1.5 SYSTEM INTERFACES

Establish system boundaries, and define major external and internal interfaces. Discuss the general flow of both execution control and data across external interfaces, including networking considerations.

7.2 System Environment

Define the environments of the system.

7.2.1 ORGANIZATIONAL ENVIRONMENT

Define the organizational environment in which the system will exist during both development and operations.

7.2.2 OPERATIONAL ENVIRONMENT

Describe the physical environment in which the system must perform, including environment characteristics and operator needs that directly affect system design. Discuss the maintenance environment as part of the operational environment.

7.2.3 USER ENVIRONMENT

Identify and describe the expected users of the system, the way in which they will be using the system, and the functional capabilities they will require when performing their activities. Interpret the term "user" to include interfacing systems and subsystems. Define the users and their needs explicitly and in such terms and detail as to make it possible to correlate system capabilities and characteristics to specific needs. (Use a cross-reference matrix if necessary.)

7.2.4 DEVELOPMENT ENVIRONMENT

Discuss the degree and type of resources, support, tools, technology, and facilities required for the system development effort.

7.3 System Operations

Describe system operations, including requirements and interfaces. Provide scenarios to explain specifics of system operations. Define personnel requirements.

7.3.1 OPERATIONAL DESCRIPTION

Specify system operational modes, contingencies, mission phases, scenarios, processing schedules, response times, and other known operational needs.

7.3.2 SIGNIFICANT OPERATIONAL REQUIREMENTS

Discuss operational requirements that will have a major impact on system design.

7.3.3 OPERATIONAL INTERFACES

Identify key operator and user positions, and highlight their capabilities. Include details of operator control and flexibility required, such as level of automation.

7.3.4 OPERATIONAL SCENARIOS

Describe scenarios of the interaction of hardware, software, operators, and users. Present assumptions about how operational tasks will be performed. Identify operational constraints.

7.3.5 PERSONNEL REQUIREMENTS

Specify skill levels required of system operators at all identified operator positions. Discuss key tradeoffs between position capabilities and skill requirements imposed on operators.

ACRONYMS

List and define all acronyms used in the project Definition.

REFERENCES

List all cited references.

Subject	Type	Standard
Current System Assessment Document	Identifier	S–3052
	Effective Date	October 1997
	Revision No.	

Approval C E Fitzgerald
CISSCO Program Director

A. PURPOSE

This standard specifies content and format requirements for the Current System Assessment Document.

B. APPLICABILITY

This standard applies to all projects that fall under the SDLCM Methodology umbrella. It is used by those persons who assess the current system as part of the Project Definition and Analysis activity within component 1 of the SDLCM Methodology.

C. REFERENCE PUBLICATIONS

The following publications contain related information:

- *SDLCM Methodology Handbook*
- SDLCM Methodology Standard S-3051, Project Definition and Analysis Document

D. STANDARD

Use the Current System Assessment Document to document the results of the assessment of the current system if the assessment is complex and cannot be summarized in the Project Definition and Analysis Document. The following paragraphs describe the content of each section.

1. INTRODUCTION

1.1 Background

Briefly describe why this assessment is being performed.

1.2 Objectives

Describe how the results of this assessment will contribute to a better understanding of the system requirements to be developed as part of the Project Definition and Analysis Document.

1.3 Scope

Describe what this product covers and what it does not cover.

1.4 Assumptions

Discuss what we have assumed because something we need to know has not been explicitly stated as a requirement or background.

1.5 Applicable Documents

List any other documents that apply.

1.6 Overview

Provide a brief overview of the sections that follow.

2. ASSESSMENT APPROACH

Describe the approach that was used to produce the assessment. For example, describe the steps that were taken as part of the assessment, such as:

- Determine the sources of information available and obtain whatever is available
- Organize and conduct interviews and focus sessions to obtain other necessary information
- Establish a template for documenting the acquired information
- Review the results from the NRC enterprise modeling effort

Describe the sources of information provided in the assessment, such as:

- User guides and documentation
- Focus sessions
- One-on-one interviews
- Surveys

Describe the tools and techniques used during the assessment, such as,

- Focus sessions
- Structured interviews

3. CURRENT ENVIRONMENT

3.1 Organizations Supporting the Environment

Describe the organizations that support the current environment and their roles.

3.2 Inventory of Current Systems

List the current systems in the environment being assessed, including such information as:

- System name
- System acronym
- System number
- Status (operational or under development)
- System purpose

3.3 Descriptions of Current Systems

3.3.1 INFORMATION SYSTEMS INTERFACE DIAGRAM

Provide a system level interface diagram showing the interfaces among existing systems or subsystems. Provide a legend to explain any symbols used in the diagram.

3.3.2 INFORMATION SYSTEMS OVERVIEW

For each system listed in Section 3.2, provide information resulting from the assessment, such as:

- System Name
- Platform
- System software
- Databases
- Programs
- Files
- Records
- Internal interfaces
- External interfaces
- Telecommunications capabilities
- Protocols

- Data usage
- Data access
- System functions

Where possible, obtain this system information from the Systems Inventory to avoid conflict, confusion, or duplication.

For each system described, describe the strengths or weakness of keeping that system or subsystem as part of the new system being envisioned in the Project Charter and being specified in the Project Definition and Analysis Document.

4. ASSESSMENT RESULTS AND RECOMMENDATIONS

While keeping the functional and data requirements in mind, summarize your assessment of the current system to determine what (if any) changes (additions, deletions, replacements) are required to fulfill the functional and data requirements of the Project Definition and Analysis. This is illustrated in Table 3052-1. Keep in mind that the goal of any system upgrade or replacement is to provide better service, faster, and less expensively than the system it is replacing. Therefore, address the issues of cost, functionality, ease of use, and efficiency.

Table 3052-1. An Assessment of the Existing and Desired Functionality or Attributes of System X

Priority	Functions or Attributes Required	Functions or Attributes that are Part of the Existing System	Additional Functions or Attributes Required
	A	A	
	B	B	
	C		C

APPENDIX A: GLOSSARY AND ACRONYM LIST

Subject	Type	Standard
Software Engineering Notebook	Identifier	S–3091
	Effective Date	November 1998
	Revision No.	1

Approval C E Fitzgerald
CISSCO Program Director

A. PURPOSE

This standard establishes the minimum requirements for software engineering notebooks (SENs).

B. APPLICABILITY

This standard applies to all software elements (for example, modules, units, data files, database subschema), whether new, adapted, or converted, composing a software system under development or in maintenance. This standard does not necessarily apply to unchanged transported software elements.

C. REFERENCE PUBLICATIONS

The following publications contain related information:

- *SDLCM Methodology Handbook*
- SDLCM Methodology Procedure P–3041, Software Engineering Notebook Processing

D. STANDARD

The SEN is an implementation workbook that consolidates the information pertinent to a software element (or set of software elements). It also ensures ready access to complete and up-to-date information for modification and auditing purposes.

Only the most current information about software development and testing is maintained in each SEN. To reduce duplication, SENs do not contain information provided in other documents or data sets. Instead the SEN provides a pointer reference to the relevant document(s). Whenever possible, the SENs should provide pointer references to on-line libraries and files rather than maintaining hard copy material.

For each project, identify a standard medium for the SENs. Suggested options include one or a combination of the following:

- On-line (either on personal computers or on the development system, if resources are available)
- Single folder or notebook for each software element

- Single folder or notebook containing all software elements for a single system, subsystem, executable image, package, or module

Include at least the following items in the SEN:

- Complete current listing(s), including prolog, program design language (PDL), unit test plan, and compiled source code
- All applicable certification records (for example, unit design, code, test, and modification)
- Most recent past listing (modified software elements only)
- For module-level notebooks, include architecture diagrams, module test plans, and associated certification records

SENs will be completed for each developed software element and updated for each modification to a software element. Software developers will maintain control of the SENs until completion of the software development life cycle (begin acceptance testing), at which time the Configuration Management Office will maintain the SEN library to ensure an accurate record of checked out and returned SENs.

Subject	Type	Procedure
Data Modeling	Identifier	P–3101
	Effective Date	November 1998
	Revision No.	1

Approval C E Fitzgerald
CISSCO Program Director

1. PURPOSE

This procedure provides direction for performing data modeling. Data modeling is used to identify the data elements that are read, transformed, or produced to satisfy the project requirements. Data modeling is performed concurrently with process modeling and together, the two modeling techniques provide the design of the solution system at the conceptual, logical, and physical levels.

2. APPLICABILITY

Data modeling is required of all projects subject to the SDLCM Methodology. However, those projects that involve only the upgrade of an existing data base to a new data base management system may require only regeneration of the physical data model (the schema and data dictionary) from the logical data model.

3. REFERENCE PUBLICATIONS

The following publications contain related information:

- *SDLCM Methodology Handbook*
- SDLCM Methodology Procedure P–3111, Process Modeling
- SDLCM Methodology Standard S–3151, Data Models
- SDLCM Methodology Standard S–3351, Data Dictionary
- SDLCM Methodology Standard S–3051, Project Definition and Analysis Document
- SDLCM Methodology Standard S–3171, Logical Design Document
- SDLCM Methodology Standard S–3172, Physical Design Document
- SDLCM Methodology Procedure P–2101, Peer Review

4. PROCEDURE

4.1 Data Flow Diagram

The data modeling procedure comprises the three major steps identified in the data flow diagram shown in Figure 3101–1.

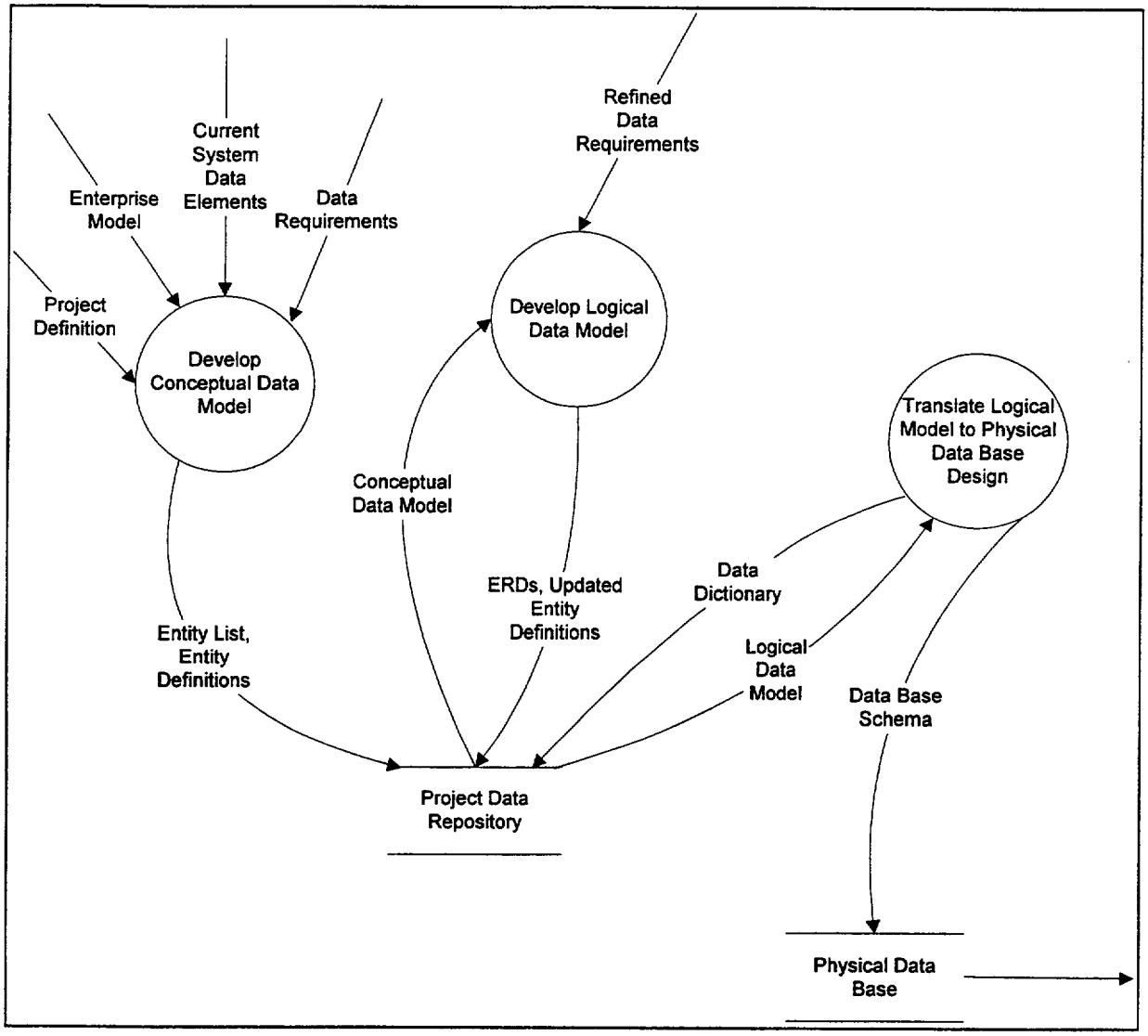

Figure 3101–1. Data Modeling Data Flow Diagram

4.2 Entry Criteria

Data modeling begins after the draft project definition is in place. The draft project definition, current system data elements, data requirements, and information from the Enterprise Model are all inputs to the procedure.

4.3 Steps

Perform the following steps to complete data modeling:

1. Develop Conceptual Data Model

- Review project definition, current system data elements, data requirements, and any information contained in the Enterprise Model applicable to the project. Identify the data entities needed to satisfy the data requirements and possible sources. If the data entities exist in the Enterprise Model, use the same names.
- Create the data entity list.
- Define each data entity. Use the guidance found in SDLCM Methodology Standard S–3151, Data Models, in setting up the definitions. Use the project's approved CASE tool to input the Conceptual Model (the data entity definitions) into the Project's data repository.
- Peer review the Conceptual Data Model following SDLCM Methodology Procedure P–2101, Peer Review.
- After the conceptual data model has been approved, place it under configuration control in the project's data repository.
- Include the Data Entity List and Definitions in the Project Definition and Analysis (PDA) Document.

2. Develop the Logical Data Model
   - Review the data requirements documented in the PDA Document and any additional information developed as a result of Component 2 activities. Retrieve the conceptual data model from the project's data repository, review, and update if necessary.
   - Determine the attributes of each data entity and add to the entity definitions.
   - Develop the system-level entity relationship diagram (ERD). Include any data exchanged with all external interfaces. Use the project's approved CASE tool to create the ERD.
   - Peer review the system-level ERD following SDLCM Methodology Procedure P–2101, Peer Review.
   - Place the approved system-level ERD under configuration control in the project's data repository.
   - Decompose the system-level entity ERD to component-level ERDs. Use the project's approved CASE tool to create the ERDs.
   - Peer review the component-level ERDs following SDLCM Methodology Procedure P–2101, Peer Review.
   - Place the approved component-level ERDs under configuration control in the project's data repository.
   - Include the Logical Data Model (ERDs, updated entity definitions) in the Logical Design Document.
3. Translate the Logical data model to the physical data base design.

- Use to project's CASE tool to convert the data definitions and ERDs to the selected data base management system's data dictionary and schema.
- Peer review the physical data dictionary and schema following SDLCM Methodology Procedure P–2101, Peer Review.
- Place the approved data base schema and data dictionary under configuration control in the physical data base under the selected data base management system.
- Include the schema as the physical data model in the Physical Design Document.

4.4 Exit Criteria

When data modeling is complete, the physical data base design is in place (that is, the physical data base is ready to be populated with actual project data) and all data needed to support the problem solution have been identified and their sources are known. The conceptual data model will have been documented in the PDA Document, the logical data model in the Logical Design Document, and the physical data model in the Physical Design Document.

4.5 Verification

Use the peer review procedure to verify the data models at each level (conceptual, logical, and physical).

4.6 Roles

Table 3101–1 is a Step-Role Table for the data modeling procedure:

Table 3101–1. Step-Role Table for Data Modeling (1 of 2)

Roles: / Steps:	Overall PM	Tech PM	Data Base Engineer	Software Designer	QA	CM
1. Develop Conceptual Data Model						
Review data-related information and identify data entities that satisfy data requirements.		S	P	S		
Create Data Entity List.		R	P			
Define each data entity.		R				
Peer Review data entity list and definitions.			S, P	P	R	
Put conceptual data model under configuration control.			S		R	P
Include conceptual data model in PDA Document.	A		P		R	
2. Develop Logical Data Model						
Review data requirements and conceptual data model.		S	P	S		
Determine data attributes and add to definitions.			P	S		
Develop system-level ERD.			P	S		
Peer Review system-level ERD			P, S	S	R	
Put system-level ERD under configuration control.			P		R	P
Develop component-level ERDs			P	S		
Peer Review component-level ERDs			P, S	S	R	
Put component-level ERDs under configuration control.			P		R	P
Include logical data model in Logical Design Document.		A	P		R	
3. Translate Logical Data Model into Physical Data Base Design.						
Use project's CASE tool to create the data dictionary and data base schema.		S	P	S		

Table 3101–1. Step-Role Table for Data Modeling (2 of 2)

Steps: / Roles:	Overall PM	Tech PM	Data Base Engineer	Software Designer	QA	CM
Peer Review physical data model			S, P	S	R	
Put physical data model under configuration control.			S		R	P
Include physical data model in Physical Design Document.		A	P		R	

Legend: P = Performs, R = Reviews, A = Approves, S = Supports

Subject	Type	Procedure
Process Modeling	Identifier	P–3111
	Effective Date	October 1997
	Revision No.	

Approval C E Fitzgerald
CISSCO Program Director

1. PURPOSE

Process modeling is used to clearly identify the activities that, when performed, will satisfy project requirements. Process modeling is performed concurrently with data modeling and together, the two modeling techniques provide the design of the solution system at the conceptual, logical, and physical levels.

2. APPLICABILITY

Process modeling is required of all projects subject to the SDLCM Methodology. However, projects that involve the packaging of commercial off-the-shelf (COTS) products may require only conceptual and logical modeling. Those that involve planned enhancement of an existing system to incorporate a new technology may only require updates of the existing logical process model and upgrade or regeneration of the physical model.

3. REFERENCE PUBLICATIONS

The following publications contain related information:

- *SDLCM Methodology Handbook*
- SDLCM Methodology Procedure P–3101, Data Modeling
- 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–3051, Project Definition and Analysis Document
- SDLCM Methodology Standard S–3171, Logical Design Document
- SDLCM Methodology Standard S–3172, Physical Design Document
- SDLCM Methodology Procedure P–2101, Peer Review

4. PROCEDURE

Conceptual process modeling is performed as a Component 1 activity to clarify the project definition and set the business problem in context for solution. Logical and physical process modeling are performed as activities of Components 3 and 4 to support the logical and physical design of the proposed solution to the business problem.

4.1 Data Flow Diagram

The process modeling procedure comprises the three major steps identified in the data flow diagram shown in Figure 3111–1.

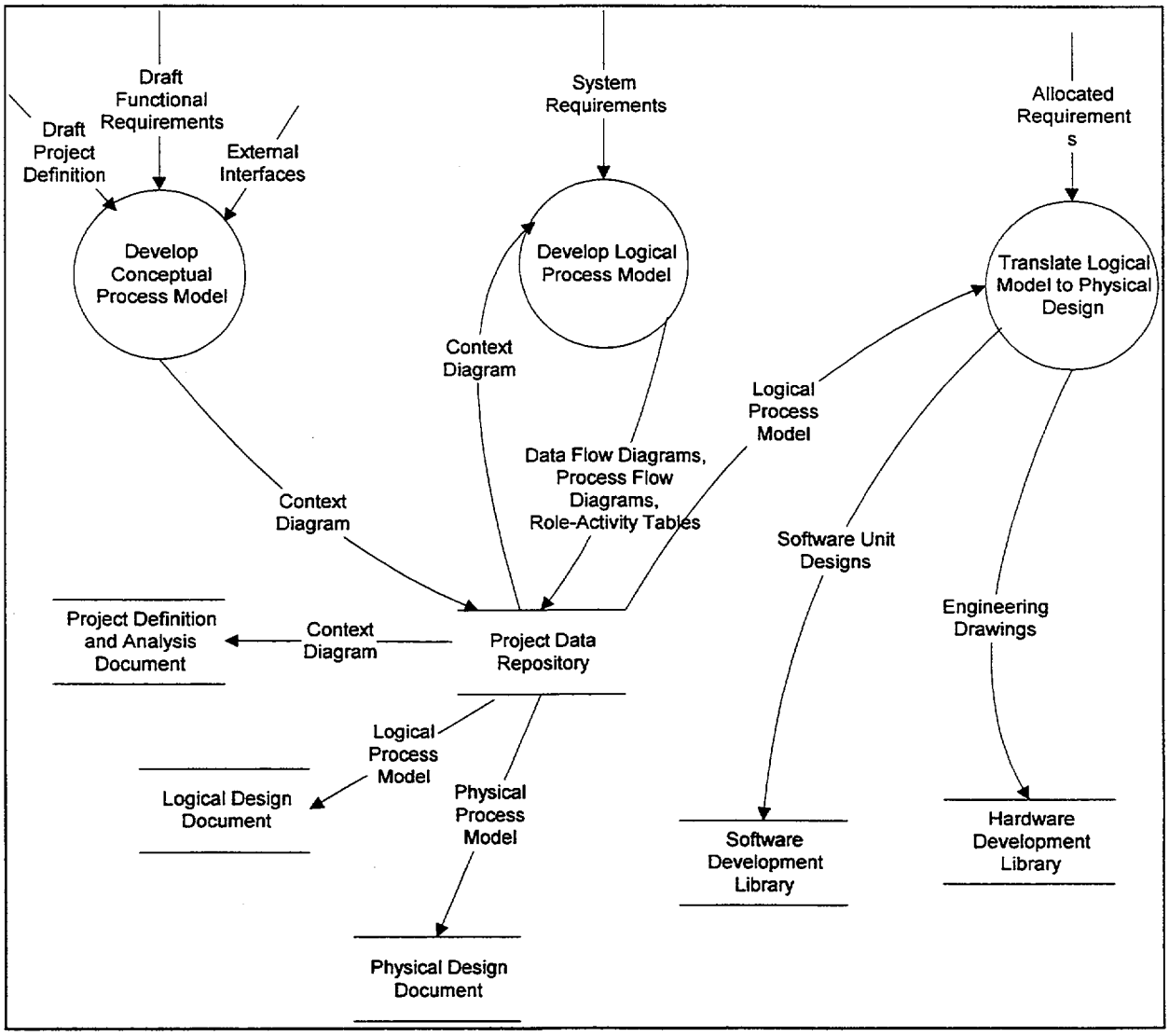

Figure 3111–1. Process Modeling Data Flow Diagram

4.2 Entry Criteria

Process modeling begins after the draft project definition is in place. The draft project definition, functional requirements, and external interface information are all inputs to the procedure.

4.3 Steps

Perform the following steps to complete process modeling:

1. Develop conceptual process model.
   - Review project definition, functional requirements that must be satisfied by the project to solve the identified business problem, and external interfaces. Identify the major functions needed to satisfy the functional requirements.
   - Develop a context diagram, following the guidance provided in SDLCM Methodology Standard S–3162, Context Diagrams. If applicable, use the project's approved CASE tool to develop the diagram.
   - Peer Review the Context Diagram following SDLCM Methodology Procedure P–2101, Peer Review.
   - After the Context Diagram has been approved, place it under configuration control in the project's data repository.
   - Include the context diagram in the Project Definition and Analysis Document.
2. Develop logical process model
   - Review system requirements documented in the PDA Document and any additional information developed as a result of Component 2 activities. Retrieve context diagram from Project's Data repository, review, and update if necessary.
   - Decompose each of the major functions into a level of activity that will provide the logical design of the selected solution to the problem. This is the system-level design.
   - Use data flow diagrams to decompose the activities that make up each major function and to depict the inputs and products of each. Follow the guidance provided in SDLCM Methodology Standard S–3163, Data Flow Diagrams, in developing data flow diagrams. If applicable, use the project's approved CASE tool to develop the data flow diagrams.

     Optionally, use process flow diagrams to depict the activities that make up each major function. Use an approved drawing tool to support creation of process flow diagrams.
   - Create a role-activity table to identify the roles responsible for performing, supporting, reviewing, and approving each major function identified. If any function is to be performed as a non-automated function, provide a role-activity table to the lowest level identified in the data flow diagrams.
   - Peer Review the Data Flow Diagrams and associated role-activity tables following SDLCM Methodology Procedure P–2101, Peer Review.
   - Place approved Process Flow Diagrams, Data Flow Diagrams and role-activity tables under configuration control in the project's data repository.

- Include the Logical Design Model (data flow diagrams, process flow diagrams, and role-activity tables) in the Logical Design Document.

3. Translate the logical process model to a physical design. For each software or hardware component:
   - Review the requirements allocated to the software or hardware component being designed and any other information that was generated during logical design activities that may impact the component. Retrieve the applicable data flow diagrams and role-activity tables from the Project's Data repository, review and update them if necessary.
   - Decompose the component into software units or hardware components. Develop the software unit-level design, documenting it in unit prologs and program design language. Document any hardware components that will be fabricated with engineering drawings.
   - Peer Review the software unit designs or engineering drawings following SDLCM Methodology Procedure P–2101, Peer Review.
   - Place approved unit designs or engineering under configuration control in the project's software or hardware library.
   - Include the Physical Design Model in the Physical Design Document.

4.4 Exit Criteria

When process modeling is complete, the design of the software or hardware components that make up the selected solution to the business problem addressed by the project will be at a level that software or hardware can be built and tested. The conceptual process model will have been documented in the PDA Document, the logical process model in the Logical Design Document, and the physical process model in the Physical Design Document.

4.5 Verification

Use the peer review procedure to verify process models at each level (conceptual, logical, and physical).

4.6 Roles

Table 3111–1 is a Step-Role Table for the process modeling procedure.

Table 3111–1. Step-Role Table for Process Modeling

Steps: / Roles:	Overall PM	Tech PM	System Analyst	Software or Hardware Designer	QA	CM
1. Develop Conceptual Process Model						
Review functional requirements and identify major functions.		S	P	S		
Develop context diagram (conceptual process model).		R	P			
Peer Review Context Diagram.		A	S	P	R	
Put conceptual process model under configuration control.			S			P
Include conceptual process model in PDA Document.	A		P		R	
2. Develop Logical Process Model						
Review system requirements and context diagram.		S	S	P		
Decompose system-level design into software or hardware design		R		P		
Develop data flow diagrams or process flow diagrams		P				
Create Role-Activity Tables		P				
Peer Review logical process model (DFDs, PFDs, and Role-Activity Tables).		A	P	S	R	
Put logical process model under configuration control.				S		P
Include logical process model in Logical Design Document.		A		P	R	
3. Translate Logical Process Model into Physical Process Model. For each component:						
Review software or hardware requirements allocated to the component.		S		P		
Decompose system-level design to software units or hardware components for each identified system component.		R		P		

Steps: / Roles:	Overall PM	Tech PM	System Analyst	Software or Hardware Designer	QA	CM
Develop software unit designs or hardware engineering drawings.				P		
Peer Review physical process model		A	P	S	R	
Put physical process model under configuration control.				S	R	P
Include physical process model in Physical Design Document.		A		P	R	

Legend: P = Performs, R = Reviews, A = Approves, S = Supports

Subject	Type	Standard
Data Models	Identifier	S–3151
	Effective Date	November 1998
	Revision No.	1

Approval C E Fitzgerald
CISSCO Program Director

A. PURPOSE

This standard specifies the format and content of data, or entity, models.

B. APPLICABILITY

This standard applies to all projects subject to the SDLCM Methodology that include software development or integration.

Members of the Development Team, designated by the Technical Project Manager, are responsible for developing and maintaining the project's data models, key managers and quality assurance personnel for reviewing them, and the Technical and Business Project Managers for approving them. The data models are made available to all members of the project team preferably in electronic form.

C. REFERENCE PUBLICATIONS

The following publications contain related information:

- *SDLCM Methodology Handbook*, Component 1
- *SDLCM Methodology Handbook*, Component 3
- *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 Procedure P–3101, Data Modeling
- SDLCM Methodology Standard S–3051, Project Definition and Analysis Document
- SDLCM Methodology Standard S–3171, Logical Design Document
- SDLCM Methodology Standard S–3172, Physical Design Document

D. STANDARD

Three levels of Data Models may be produced by a project following the SDLCM Methodology:

- The **Conceptual Data Model**, also known as the scope-setting version of the logical data model, identifies groupings of data important to the business situation that the

project addresses. This data model is documented in the Project Definition and Analysis (PDA) Document.

- The **Logical Data Model**, completed version, presents a fully attributed and normalized data model and identifies the relationships among all data entities. This data model is documented in Logical Design Document
- The **Physical Data Model**, the design model, describes how data will be distributed to different processing nodes and structured to meet performance objectives in a specific physical implementation. This data model is documented in the Physical Design Document.

Table 3151–1 identifies the items that make up each of the three data models:

Table 3151–1. Data Models

Conceptual Model	Entity List Entity Definition
Logical Model	Entity List Entity Definition Entity Relationship Diagram (ERD) Attribute Attribute Definition Information Type (Domain)
Physical Model	Data Catalog Relational Schema Relational Table Structure Diagram

Refer to the *Systems Development CASE Tool Guidelines* and to SDLCM Methodology Procedure P–3101, Data Modeling, for descriptions of the processes used to create and refine data models.

D.1 Definitions

An *entity* describes a person, place, thing, or event about which data is stored. It may refer to a tangible object in the real world, such as CUSTOMER or PRODUCT, or it may refer to an intangible business concept, such as ASSIGNMENT or CLAIM.

A *relationship* describes the connection between two entities. For example, "EMPLOYEE **is assigned to** PROJECT" identifies two entities, EMPLOYEE and PROJECT, and describes their relationship, **is assigned to**. The relationship is usually expressed in the form of a verb.

Cardinality describes the number of occurrences of a first entity that may exist in a relationship with a second entity. The possible types of cardinality are one-to-one, one-to-many, and many-to-many.

Entity *subtypes* are used to clarify otherwise confusing relationships. For example, BOAT and AUTO may be subtypes of an entity VEHICLE that can be either a BOAT or an AUTO. Each subtype has at least one unique relationship with another entity. For example, only a BOAT **is moored at** a DOCK.

An *attribute* is a data element that is assigned as a characteristic of a specific entity. As an example, the attributes of a data entity EMPLOYEE may include the employee number, name, address, job code, and department.

Two special types of attributes are the primary key and the foreign key. A *primary key* is an attribute or set of attributes that uniquely identifies an entity. In the example above, the employee number is the primary key. A foreign key is an attribute of one entity that is part of a primary key of another entity.

D.2 Conceptual Data Model

Create the conceptual data model to identify and briefly describe the high-level entities that are within the scope of the project. Note: Use the conceptual data model as the starting point for a more detailed definition of project data requirements.

D.2.1 ENTITY LIST

Identify and list only the high-level data entities important to the project. Ensure that each listed entity is independent of any physical constraints; do not reflect the constraints of any database or data file.

Note: The entity list, generated as an activity of Component 1 (Define Initial Project Requirements) of the SDLCM Methodology, is based on partial understanding and will be refined as a data modeling activity of Component 3, Design the Solution.

Refer to Section 3.1, Data Objects, of the *Standards and Conventions* document for the conventions used to name data entities.

D.2.2 ENTITY DEFINITION

Provide as much of the following information as is available to define each entity in the entity list:

- Name
- Identifier (primary key)
- List of included subtypes, if any
- Description (several sentences)

- Average volume at conversion
- Annual growth percentage
- Active life
- Estimated size in characters (preliminary)

Refer to Appendix C, Guidelines for Object Definitions, of the *Standards and Conventions* document for guidelines in defining data entities.

D.3 Logical Data Model

Create the logical data model to identify all of the entities needed by the project, their attributes, relationships, and the domains, or information types, that specify the format and value sets of the attributes.

D.3.1 ENTITY LIST

Refine the entity list from the conceptual data model to include all data entities needed by the project. Ensure that each listed entity is independent of any physical constraints; do not reflect the constraints of any database or data file.

Refer to Section 3.1, Data Objects, of the *Standards and Conventions* document for the conventions used to name data entities.

D.3.2 ENTITY DEFINITION

Include the following information to define each entity in the entity list:

- Name
- Short Name
- Identifier (primary key)
- Description (several sentences)
- List of included attributes
- Average volume at conversion
- Annual growth percentage
- Active life
- Archived life
- Estimated size in characters
- Aliases (use with caution, if at all)

Refer to Appendix C, Guidelines for Object Definitions, of the *Standards and Conventions* document for guidelines in defining data entities.

D.3.3 ENTITY RELATIONSHIP DIAGRAM

Develop one or more entity relationship diagrams (ERDs) that show the project's entities and their relationships. Ensure that the ERD:

- Shows all entities (although not necessarily on one diagram)
- Resolves all many-to-many relationships into pairs of one-to-many relationships
- Identifies and defines all entity attributes of interest to the business process
- Removes subtypes, if applicable
- Identifies foreign keys, if applicable
- Is in at least third normal form (but may include derived data attributes)
- Identifies attribute domains
- Is independent of physical constraints such as those imposed by the target DBMS
- Is limited to the scope of the project

Refer to Section 3.2.12, Entity Relationship Diagram Graphical Conventions, of the *Systems Development CASE Tool Guidelines* for guidance in creating the actual ERD.

D.3.4 ATTRIBUTE AND ATTRIBUTE DEFINITION

Document the attributes of each entity. Briefly define each attribute by identifying its optionality and uniqueness. Refer to Section 3.1.6, Attribute, of the *Standards and Conventions* document for the conventions used to name and define data attributes.

Refer to SDLCM Methodology Procedure P–3101, Data Modeling, and Section 3.2.5, Attributes, of the *Systems Development CASE Tool Guidelines* for the process involved in identifying and refining the attributes selected for the data entities.

D.3.5 ATTRIBUTE DOMAIN

Specify the characteristics that portray the structure of the attribute and define how it will be stored physically. Include the physical characteristics, such as length and character type information, about each attribute. Identify the domains of allowable values of the attribute, such as days of the week or months of the year, and the range of values.

Refer to Section 3.1.8, Information Type, of the *Standards and Conventions* document for the conventions used to characterize attributes.

D.4 Physical Data Model

Create the physical data model to depict all aspects of how data will be stored in the computer system.

D.4.1 DATA CATALOG

Document the results of translating the data entity definitions to a baseline data element model.

Each object (entity type, attribute type, relationship type, information type, and subtype sets) in the logical data model relates to a corresponding object in the physical model.

Refer to Section 4.2.1, Translation to the Data Catalog, of the *Systems Development CASE Tool Guidelines* for further information on the data catalog.

D.4.2 RELATIONAL SCHEMA

Provide a relational schema diagram to depict the relational tables derived from the ERD and the relationships between the tables.

Refer to Section 4.1.1, Relational Schema, of the *Standards and Conventions* document for the conventions used to document the schema.

D.4.3 RELATIONAL TABLE STRUCTURE DIAGRAM

Provide a relational table structure diagram for each entity type contained in the data catalog.

Refer to Section 4.3.1.2, Data Structure, of the *Systems Development CASE Tool Guidelines* for information on how the translation is accomplished.

Refer to Section 4.1.2, Relational Table, of the *Standards and Conventions* document for the conventions used to document the relational table structure.

Subject	Type	Standard
Process Models	Identifier	S–3161
	Effective Date	November 1998
	Revision No.	1

Approval C E Fitzgerald
CISSCO Program Director

A. PURPOSE

This standard defines the format and content of process models.

B. APPLICABILITY

This standard applies to all projects subject to the SDLCM Methodology that include software development or integration.

Members of the Development Team, designated by the Technical Project Manager, are responsible for developing and maintaining the project's process models, key managers and quality assurance personnel for reviewing them, and the Technical and Business Project Managers for approving them. The process models are made available to all members of the project team, preferably in electronic form.

C. REFERENCE PUBLICATIONS

The following publications contain related information:

- *SDLCM Methodology Handbook*, Component 1
- *SDLCM Methodology Handbook*, Component 3
- *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 Procedure P–3111, Process Modeling
- SDLCM Methodology Standard S–3163, Data Flow Diagrams
- SDLCM Methodology Standard S–3162, Context Diagrams
- SDLCM Methodology Standard S–3051, Project Definition and Analysis Document
- SDLCM Methodology Standard S–3171, Logical Design Document
- SDLCM Methodology Standard S–3172, Physical Design Document

D. STANDARD

Two levels of Process Models may be produced by a project following the SDLCM Methodology:

- The **Conceptual Process Model**, also known as the scope-setting version of the logical process model, is called a context diagram. This process model is documented in the Project Definition and Analysis (PDA) Document.
- The **Logical Process Model**, completed version, provides all the detail required to effectively document and communicate to the business community the project team's understanding of the functional requirements of the application. The logical process model includes data flow diagrams (DFDs) and process definitions. This process model is documented in Logical Design Document

Refer to the Section 3.3, Process Model, of the *Systems Development CASE Tool Guidelines* and to SDLCM Methodology Procedure P–3111, Process Modeling, for descriptions the processes used to create and refine process models.

D.1 Conceptual Process Model, the Context Diagram

The context diagram gives a visual representation of the functional scope of the project. Include the context diagram in the Project Definition and Analysis (PDA) Document and in the Logical Design Document.

Refer to SDLCM Methodology Standard S–3162, Context Diagrams, for the format and content of this process model.

D.2 Logical Process Model

Create an initial logical process model that effectively documents and communicates to the business community the project team's understanding of the functional requirements of the application. Include all of the processes required to fully depict the business problem being solved; do not differentiate between processes that are automated or manual and those that may be converted from manual to automated.

Refine the logical process model throughout the design process. Refer to SDLCM Procedure P–3111, Process Modeling, or Section 3.3, Process Model, of the *Systems Development CASE Tool Guidelines* for guidance in refining the logical process model.

Include the logical process model in the Logical Design Document.

D.2.1 DATA FLOW DIAGRAMS

Provide DFDs to show how processes use data and to decompose the context diagram to the lowest meaningful level that reflects the structure of the application.

In the first level DFD, indicate the major components or subsystems of the proposed solution to the business problem. Include both automated and manual processes and do not differentiate between them.

Refer to SDLCM Methodology Standard S–3163, Data Flow Diagrams, for the format and content of DFDs.

D.2.2 PROCESS DEFINITION

List and define each process object identified in the DFDs. Process objects include process (or function) names, data flows, data stores, and external agents (data sources and sinks).

Refer to Section 3.2, Process Objects, of the *Standards and Conventions* document for the conventions used to define each type of process object.

Subject	Type	Standard
Context Diagrams	Identifier	S–3162
	Effective Date	November 1998
	Revision No.	1

Approval C E Fitzgerald
CISSCO Program Director

A. PURPOSE

This standard specifies the format and content of context diagrams.

B. APPLICABILITY

A context diagram is prepared and maintained for all projects that are developed using the SDLCM Methodology. It is initially developed as an activity of Component 1, Define Initial Project Requirements. The context diagram is refined, and may be redrawn, as an activity of Component 3, Design the Solution.

The Technical Project Manager is responsible for developing and maintaining the context diagram and key managers (especially the Business Project Manger and Business Advocate) and quality assurance personnel for reviewing it. The Executive Sponsor approves the context diagram that is produced as an activity of Component 1; the Overall Project Manager approves any updates to it. The context diagram is made available to all members of the project team, preferably in electronic form.

C. REFERENCE PUBLICATIONS

The following publications contain related information:

- *SDLCM Methodology Handbook*, Component 1
- *SDLCM Methodology Handbook*, Component 3
- *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–3051, Project Definition and Analysis Document
- SDLCM Methodology Standard S–3171, Logical Design Document
- SDLCM Methodology Standard S–3151, Data Models
- SDLCM Methodology Standard S–3163, Data Flow Diagrams

D. STANDARD

A context diagram is a special top-level data flow diagram (DFD) that names the system to be developed or modified and that defines the bounds of a system in terms of the data it receives and generates. A context diagram provides a visual representation of the functional scope of the project.

The context diagram is initially produced as an activity of Component 1, Define Initial Project Requirements, and is included in the Project Definition and Analysis (PDA) Document. It is updated, if necessary, during the data modeling activities of Component 3, Design the Solution, and is included in the Logical Design Document.

Refer to Section 2.3, Context Diagram, of the *Systems Development CASE Tool Guidelines* for additional information on context diagrams.

D.1 Context Diagram Content

Context diagrams convey the names (at a high level) of the information received and generated by the system to be developed or modified. The sources and targets of the information are represented as external agents on the context diagram. The external agents can be NRC organizations and existing systems or processes outside of the application's scope. The process name (statement) is a noun phrase that names the system to be developed or modified. Follow the guidance in Section 3.2, Process Objects, of the *Standards and Conventions* document for process and agent names.

D.2 Context Diagram Format

Represent a context diagram as a DFD with a single, large unnumbered process symbol (circle) that contains the name of application to be developed by the project. Data flows (arrows) on the left of the process symbol normally identify the external data that will be received during system operations; data flows on the right normally identify the major data to be generated.

Represent the information agents as sources and sinks. (See SDLCM Methodology Standard S–3163, Data Flow Diagrams, for guidance in representing sources and sinks.) Use sources and sinks to identify the external systems that will provide data to or receive data from the system to be developed or modified. An example context diagram is shown in Figure 3162–1.

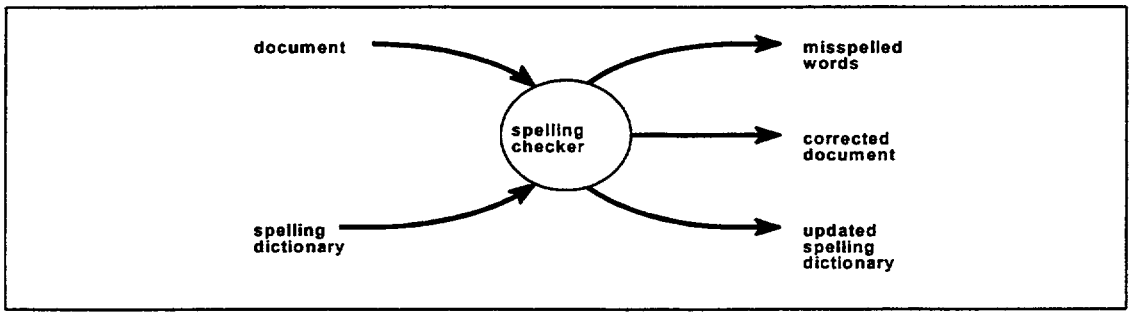

Figure 3162–1. Sample Context Diagram

Subject	Type	Standard
Data Flow Diagrams	Identifier	S–3163
	Effective Date	October 1997
	Revision No.	

Approval _____
CISSCO Program Director

A. PURPOSE

This standard defines the format and content of data flow diagrams (DFDs).

B. APPLICABILITY

This standard applies to all DFDs produced by projects subject to the SDLCM Methodology. A project produces DFDs as part of defining system requirements using structured requirements definition techniques. The Context Diagram, the top-level DFD is produced and included in the Project Definition and Analysis (PDA) Document as the output of an activity of Component 1, Define Initial Requirements, of the SDLCM Methodology. As requirements definition and analysis continue as activities of Component 3, Design the Solution, the remaining DFDs are developed and are included in the Logical Design Document.

This standard is used by Development team members responsible for requirements definition to produce the DFDs, and the Technical Project Manager and Quality Assurance personnel for reviewing them.

The DFDs are living work products of requirements definition and analysis activities and are maintained for the life of the project. The DFDs are made available to all project personnel, preferably in electronic form.

C. REFERENCE PUBLICATIONS

The following publications contain related information:

- *SDLCM Methodology Handbook*, Component 1
- *SDLCM Methodology Handbook*, Component 3
- *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–3162, Context Diagrams
- SDLCM Methodology Standard S–3161, Process Models
- SDLCM Methodology Procedure P–3111, Process Modeling

D. STANDARD

DFDs are hierarchical and logical graphical representations of data that are processed and generated during system operations. Create DFDs as part of the requirements definition and analysis process to show how the project's processes use data and to decompose the context diagram to the lowest meaningful level that reflects the structure of the application.

Refer to the Section 3.3.2, Data Flow Diagrams, of the *Systems Development CASE Tool Guidelines* and to SDLCM Methodology Procedure P–3111, Process Modeling, for descriptions the processes used to create and refine process models.

Refer to Section 3.2, Process Objects, of the *Standards and Conventions* document for the conventions used to name each type of process object.

Subject	Type	Standard
External Systems Interface Diagrams	Identifier	S–3164
	Effective Date	December 1999
	Revision No.	1

Approval C E Fitzgerald
CISSCO Program Director

A. PURPOSE

This standard specifies the format and content of an external systems interface diagram.

B. APPLICABILITY

An external systems interface diagram is prepared and maintained for all projects that are developed using the SDLCM Methodology. It is developed as an output of Component 3, Design the Solution. It takes into consideration information contained in the Project Definition and Analysis Document and the Logical Design Document. If needed, it may be embedded within the Physical Design Document. For transition of a legacy system, an external systems interface diagram may alternatively be embedded within the As-Built System Documentation as a product of Component 6.

The Technical Project Manager and members of the Development Team responsible for system or software architecture use this standard for documenting the interfaces between the system under development and external agents or systems.

C. REFERENCE PUBLICATIONS

The following publications contain related information:

- *SDLCM Methodology Handbook*, Component 1
- *SDLCM Methodology Handbook*, Component 3
- *SDLCM Methodology Handbook*, Component 6
- *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–3051, Project Definition and Analysis Document
- SDLCM Methodology Standard S–3151, Data Models
- SDLCM Methodology Standard S–3163, Data Flow Diagrams
- SDLCM Methodology Standard S–3171, Logical Design Document
- SDLCM Methodology Standard S–3172, Physical Design Document

- SDLCM Methodology Standard S–4151, As-Built System Documentation

D. STANDARD

An external systems interface diagram is a high-level data flow diagram that shows an application with its interfaces to existing or new computer systems or agents. It has these object types:

- **Process.** All processes within the scope of the application release are included in one central process, whose name is the same as the application.
- **External agents.** External agents are outside the scope of the application but interface with it in some way. An external agent could be another system within the enterprise, an external regulatory agency, or another company, such as a customer or a supplier.
- **External interfaces.** These are data flows between an external agent and the enterprise. They may be implemented as files, messages, parameters, or a shared database with an external agent.
- **Shared external databases.** This is a special case of an interface with an external agent using a common database that is outside the application.

Interfaces are shown as either:

- **Interfaces** (data flows) between the system and an external agent representing an external system
- **Shared external databases** referenced by both the new system and the external agent

An example of an external systems interface diagram is shown in Figure 3164–1.

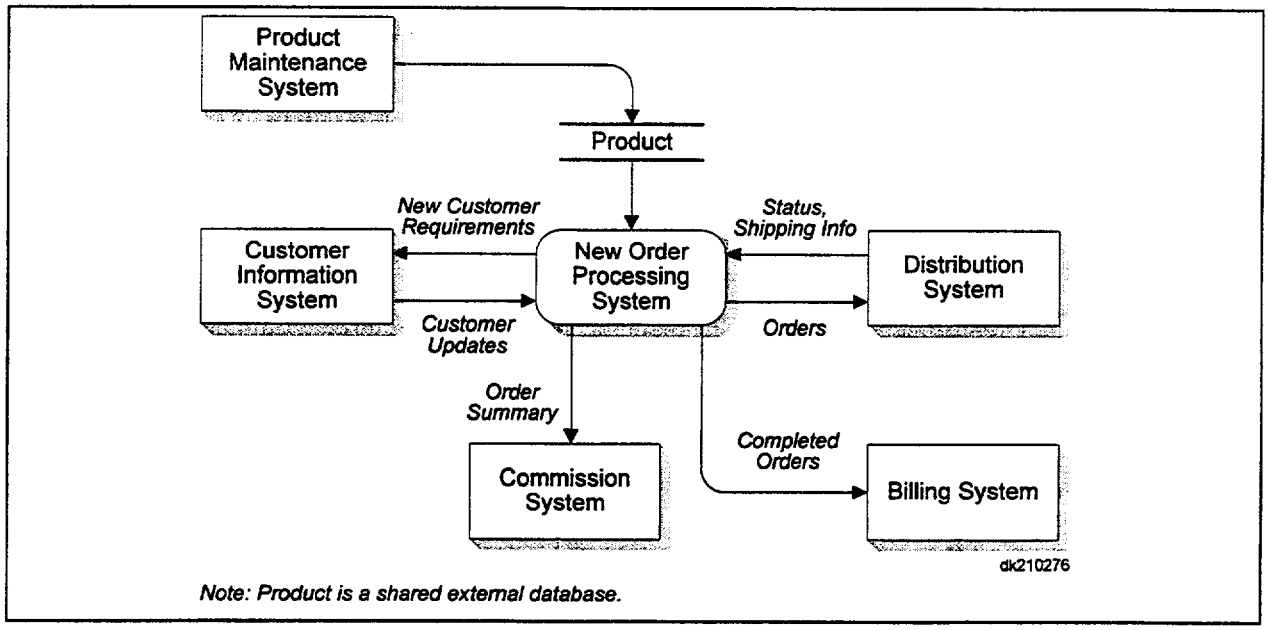

Figure 3164–1. Sample External Systems Interface Diagram

Subject	Type	Standard
Logical Design Document	Identifier	S–3171
	Effective Date	November 1998
	Revision No.	1

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; 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.

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–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–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 Project Definition and Analysis (PDA) Document, into the functions to be performed by the hardware, software, and firmware components of the system. 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 and delete sections and subsections that are not applicable.

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

Briefly describe the purpose of the selected solution's logical design. Using the PDA Document as a starting point, describe the problem to be solved, identify the system being designed, and specify the applicable references to other project documents.

1.1 Background

Identify the problem to be solved by the system under design. Refer to the PDA Document and expand on the information contained there as needed.

1.2 Objectives

Specify the business objectives the system is to support, the critical functions it should achieve, and the quantifiable criteria it must meet to succeed. Refer to the project's objectives contained in the PDA Document 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 PDA Document 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 PDA Document 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.

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.

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 requirement to its source in the system requirements specification (SRS), documented in the PDA Document.

APPENDICES (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.

Subject	Type	Standard
Physical Design Document	Identifier	S–3172
	Effective Date	November 1998
	Revision No.	1

Approval C E Fitzgerald
CISSCO Program Director

A. PURPOSE

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

B. APPLICABILITY

A Physical 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 physical solution; key managers and quality assurance personnel use it for reviewing the physical design; and the Executive Sponsor uses it when approving the Physical Design Document. The Physical Design Document is made available to all members of the project team, preferably in electronic form.

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–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–3351, Data Dictionary
- SDLCM Methodology Standard S–3171, Logical Design Document

D. STANDARD

The Physical Design Document summarizes the results of translating the logical design objects into physical design objects. The physical design objects include a relational schema, a relational

table structure diagram, the beginnings of the data definition language (DDL), a Data Dictionary, and the screen prototypes.

Tailor this standard as needed to be consistent with the size, scope, and complexity of the system. Add sections and subsections for special topics and delete sections and subsections that are not applicable.

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

1. INTRODUCTION

1.1 Background

Describe the design activities that preceded the creation of this Physical Design Document. In particular, briefly describe the activities that were performed to translate the logical design to the physical design and what computer-aided software engineering (CASE) tools were used, if any.

1.2 Objectives

Describe the objectives of creating this Physical Design Document and briefly describe how it will be used as input into the Engineer the Solution Component.

1.3 Scope

Describe what the Physical Design Document covers and what it does not cover.

1.4 Assumptions

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

1.5 Applicable Documents

List any documents that apply.

1.6 Overview

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

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

2. APPROACH

Describe how the physical design was created from the logical design. Include descriptions and results from studies, modeling activities, or simulations. If applicable, include the details and results of commercial off-the-shelf (COTS) product evaluations. 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.

3. PHYSICAL MODEL

3.1 System Architecture

3.1.1 SYSTEM COMPONENTS

Describe the physical characteristics of the network, client-server, or desktop environment on which the application system will be deployed. Include discussion of environmental constraints in terms of external communications, product compatibility or limitations, and any other technical issues.

For network architectures, characterize the network with respect to single- or multi-user capabilities and client-server architecture. Describe the network topology, connectivity, and protocols if they are significant considerations for the physical design. If applicable, describe physical security measures that are provided.

For client-server architectures, describe the physical characteristics of the file server system. Describe the different types of clients, and provide the configurations for each in terms of hardware and software. Identify any modifications or enhancements required or recommended.

Describe any additional servers such as those used for application and database services. For each additional server, describe its interface to the network and its security processes.

Describe printers, scanners, and other peripheral or physical components that will be used with the application system.

3.1.2 SOFTWARE COMPONENTS

Describe the software components of the system and the functions that each will provide.

For each COTS product, indicate if the product will be used as-is to satisfy part or all of a requirement (or requirements), or describe how the products must be customized or modified. Customization or modification method include:

- Use configuration capabilities of the COTS product
- Use the Application Programming Interface (API) kits of the COTS products
- Create new functionality through the use of COTS programming tools
- Add and integrate additional COTS products

Use structure charts or object-oriented diagrams expanded to the unit level, showing data flow, control, input, and output.

Describe the external interfaces of the software components; that is, the application interfaces for interprocess communication or data sharing that must be considered when replacing or upgrading a component.

3.1.3 USER INTERFACE

Briefly describe how a user will access and interface with the application system. For example, discuss the use of a common user interface, if applicable, to integrate desktop applications for word processing, spreadsheets, electronic mail, and database applications. Describe the types of security restrictions that will be in place at the user interface.

Provide detailed descriptions of processing initiated by operator-specified input and actions in terms of control points, functions performed, and results obtained. Provide a listing of numbered error messages with descriptions of system and user actions.

Describe critical components of the interfaces such as the need for synchronization of operations (for example, data read and write) by either user data entry or automatic means.

3.1.4 EXTERNAL INTERFACES

Provide a description of each external interface (for example, interfaces to the network topology, another system, subsystem, or configuration item). 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.

3.1.5 INTERNAL INTERFACES

Provide a description of each interface between and among the components of a configuration item. 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.

3.2 Physical Data Model

3.2.1 GENERAL MAPPING OF THE LOGICAL MODEL OBJECTS TO THE PHYSICAL MODEL OBJECTS

Describe how the logical model was translated to the physical model. Describe any intermediate data structures that were generated and how they were generated. For example, if a tool was used, specify what tool and version of that tool was used.

For example, Table 3172–1 shows how the Logical Model Objects were translated to Physical Model Objects—the relational schema.

Table 3172–1. Logical to Physical Model Object Comparison

Logical Design Logical Model Objects	Physical Design Relational Schema
Entity Types	Relational Tables
Attribute Types	Columns
Relationship Types	Relational Schema Relationships
Information Types	Data Types
Subtype Set	Data Subtypes

3.2.2 SPECIFIC MAPPING OF THE LOGICAL MODEL OBJECTS TO THE RELATIONAL SCHEMA

List the relational schemas and their components, including:

- Relational tables that represent each entity
- Columns that represent attributes
- Relational schema relationships
- Database primary, foreign, and secondary keys
- Data types

Show how the relational schema components map back to the logical design using tables such as those shown in Tables 3172–2 through Table 3172–4.

Table 3172–2. Mapping of Entity and Attribute Types to Relational Tables and Columns

Logical Design Entity and Attribute Types	Physical Design Relational Tables and Columns
Entity Type 1	Relational Table 1
• Attribute Type A	• Column A, Primary Key
• Attribute Type B	• Column B
Entity Type 2	Relational Table 2
• Attribute Type C	• Column C, Primary Key
• Attribute Type D	• Column D
• Attribute Type E	• Column E
Entity Type *n*	Relational Table *n*

Table 3172–3. Mapping of Relationship Types to Relational Schema Relationships

Logical Design Relationship Types	Physical Design Relational Schema Relationships
Relationship Type 1	Relational Schema Relationship 1
Relationship Type 2	Relational Schema Relationship 2
Relationship Type *n*	Relational Schema Relationship *n*

Table 3172–4. Mapping of Information Types to Data Types

Logical Design Information Types and Subtype Sets	Physical Design Data Types and Data Subtypes
Information Type 1	Data Type 1
• Subtype Set A	• Data Subtype A
• Subtype Set B	• Data Subtype B
Information Type 2	Data Type 2
• Subtype Set C	• Data Subtype C
• Subtype Set D	• Data Subtype D
• Subtype Set E	• Data Subtype E
Information Type *n*	Data Type *n*

3.2.3 COMPLETING THE PHYSICAL DESIGN

Describe how the physical design was completed after the logical model was translated into the physical model. Include:

- Views specified

- Indexes identified
- Data structures that have been redefined for efficiency
- Column names that have been changed for clarity

3.2.4 DATA DEFINITION LANGUAGE

List the Data Definition Language (DDL) that was generated from the relational schema. Include data dictionary information, such as:

- Tables or view of tables and their definitions
- Columns for the tables or views showing:
  - ◊ Column names
  - ◊ The column's data type format and length
  - ◊ The column's definition
  - ◊ The column's data type set definition
  - ◊ The column's data type set comments

3.3 Physical Process Model

Document the process models that evolved from the logical design, including:

- Process Flow Diagrams
- Screen prototypes

3.3.1 PROCESS FLOW DIAGRAMS

Use process flow diagrams (See in SDLCM Methodology Procedure P–3111, Process Modeling) to document all automated processes. Expound on ideas from the logical model's data flow diagrams and process definitions.

3.3.2 SCREEN PROTOTYPES

Document the screen prototypes that evolved from the logical model's:

- Data flow diagrams
- Process Definitions
- Data flow view

For each screen, specify:

- Input fields
- Processing
- Output fields
- Dialog boxes
- Triggers

3.3.2.1 Screen Prototype 1

3.3.2.2 Screen Prototype 2

3.3.2.*N* Screen Prototype *N*

APPENDIX A. REQUIREMENTS-TO-DESIGN TRACEABILITY

Update the requirements-to-design traceability matrix, as appropriate, and either include a copy or refer to the on-line traceability matrix. For each design element, indicate whether requirements are to be satisfied using COTS products, newly developed code, or reused code.

APPENDIX B. DATA DICTIONARY

Provide a copy of the project's data dictionary or refer to the project's on-line data model and data dictionary.

APPENDIX C. COTS PRODUCTS EVALUATION (IF NEEDED)

If a COTS products evaluation is included:

- List any constraints and assumptions that will be applied to COTS evaluation and selection.
- If performing an industry survey, characterize the industry with regard to the target product category, including information about vendors. Describe the technologies for the target product category, providing basic definitions for fundamental terminology. Describe in general terms what products are available, what they do, and how they are applied.
- Include descriptions for recommended or required supporting hardware and software.
- Briefly discuss government and commercial standards, regulations, and laws that apply to the processes or the hardware and software.
- Describe the basis for selection of candidate products (for example, price range, vendor reputation, platforms supported). Where possible, group product types together and describe the distinguishing characteristics of each group. Briefly describe each candidate COTS product.
- List evaluation criteria for COTS product selection with their corresponding weights. Criteria may include the following (modify the list as applicable): capabilities provided, constraints met, cost-benefit ratio, platforms and applications supported, performance, scalability for NRC's current and future needs, compatibility with other system components, proven track record or vendor credibility, price, licensing costs and flexibility, and configuration possibilities.

- Provide an evaluation of the COTS products, mapping each product's capabilities to the functional requirements of the system and identifying where customization or additional software or hardware would be necessary.
- Select the COTS product that is most functional, technically, and economically suitable for fulfilling the NRC's requirements. State the reasons for the selection, including the weighing of major tradeoffs between or among products. Specify the additional development work that will be required or additional products that would have to be purchased to support the selected products or products.
- Where possible, provide detailed information about the top two or three contender products as collected from vender literature and third-party reviews. Include vendor addresses, contact names and telephone numbers, and, if applicable, web site information.

APPENDICES (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 Physical Design Document.

REFERENCES

List all cited references.

Subject	Type	Standard
Data Dictionary	Identifier	S–3351
	Effective Date	October 1997
	Revision No.	

Approval C E Fitzgerald
CISSCO Program Director

A. PURPOSE

This standard specifies the content and format requirements for a Data Dictionary.

B. APPLICABILITY

This standard applies to any Data Dictionary produced by a project subject to the SDLCM Methodology. A project produces a Data Dictionary as part of designing and implementing the database(s) used by the system under development.

This standard is used by Development team members responsible for data modeling and database design to produce the Data Dictionary. It is also used by the Technical Project Manager and Quality Assurance personnel for reviewing the project's Data Dictionary.

The Data Dictionary is a living work product of detailed data and software design and implementation activities and is maintained for the life of the project. The Data Dictionary is made available to all project personnel, preferably in electronic form.

C. REFERENCE PUBLICATIONS

The following publications contain related information:

- *SDLCM Methodology Handbook*, Component 3
- *SDLCM Methodology Handbook*, Component 4
- *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–3172, Physical Design Document
- SDLCM Methodology Standard S–3151, Data Models
- SDLCM Methodology Procedure P–3101, Data Modeling

D. STANDARD

Refer to Sections 4.3 and 4.4 of the *Systems Development CASE Tool Guidelines* and to SDLCM Methodology Procedure P–3101, Data Modeling, for descriptions the processes used to develop and evaluate the Data Dictionary.

Refer to Sections 3.1 and 4.1 of the *Standards and Conventions* document for the conventions used to name each type of data object.

D.1 Data Dictionary Definition

A data dictionary is a mechanism for defining the data elements identified in structured analysis and structured design products (DFDs, functional specifications, or the dictionary itself). It is a place for users, designers, programmers, and testers to determine what constitutes data flows, data stores, and structure chart data couples; to look up unfamiliar terms; and to review data requirements.

To provide current and accurate information during development, establish one data dictionary on the project that is accessible to all personnel. Establish a means of controlling changes such that all users are aware of them and that the same entry cannot be changed simultaneously by several people.

Because of the amount of work necessary to maintain a large data dictionary, the use of automated tools is strongly recommended.

D.2 Data Dictionary Contents

Define the following items in a data dictionary:

- Each unique data flow or unique data couple and component included in a data flow
- Each unique data store or unique file and component included in a data store
- Anything else in the data flow diagrams (DFDs) or software architecture diagrams that needs a definition (for example, terms, abbreviations, or acronyms that are not part of a typical user's vocabulary)

Each entry in a data dictionary contains the name of the item being defined, all aliases by which it is known, its definition, and any notes or comments to further explain the item.

D.2.1 ITEM NAME

Include the name of the item as it appears in the DFD, software architecture diagram, or elsewhere in the data dictionary.

D.2.2 ALIASES

Items have aliases when they are known by two or more names. Avoid the use of aliases in the DFDs and software architecture diagrams, but when they do occur, give each its own entry in the

data dictionary. List all aliases by which the entry is known. Do not include a redundant definition for each alias. Each alias should point to a single entry that contains the full definition.

D.2.3 DEFINITION

Each entry in the data dictionary must have a definition, which may be textual or composite.

D.2.3.1 Data Elements

Data elements are those items that do not need to be or cannot be further decomposed. Define data elements in the dictionary by their values and their corresponding meanings. To avoid introducing redundancy with functional specifications, do not include information about processing in the definition associated with the data element.

D.2.3.2 Composite Data

Composite data is composed of logical combinations of data elements or smaller divisions of composite data. Each item within the definition must appear as an entry in the data dictionary.

Separate the name from the definition with an equal sign (=), which is the shorthand notation for "is composed of."

Define a composite data element in terms of the following:

- Sequence (concatenation) of data elements: Use a plus sign (+) as the symbol for a logical AND.
- Repeated data elements: Indicate repeated elements or element sets by enclosing them in braces { }. Indicate a set number of iterations by placing numbers before and after the braces 1{...}5. The numbers indicate the lower and upper limits of the iteration.
- Selection from a set of data elements: Indicate a logical OR by a vertical line (|), and an EITHER .. OR by enclosing the choices in square brackets and separating them with the logical OR symbol. Any number of choices may be enclosed in the brackets [...|...|...]. Optional elements or composites are enclosed in parentheses ( ).
- Combinations of the above: The above symbols may be combined in any order to describe the item in the dictionary. An example of a composite data flow might be flow = a + b + [c | d] + 1{ e }5

D.2.4 NOTES

When relevant, apply notes to the data dictionary entry. Items that might appear in the notes field are

- Assumptions
- Response times
- Concerns
- Organization of data stores

- Access requirements
- Conditions that generate the defined control signal

Indicate comments in a data dictionary by enclosing them in asterisks (*).

Product Engineering
Series 4000

Subject	Type	Standard
As-Built System Documentation	Identifier	S–4151
	Effective Date	December 1999
	Revision No.	

Approval _____
CISSCO Program Director

A. PURPOSE

This standard specifies the content and format requirements for As-Built System Documentation.

B. APPLICABILITY

This standard applies *only if* NRC or a contractor has developed an application system without the benefits that would have been derived by following the guidance of NRC's SDLCM Methodology and if that system is now being transitioned to a different NRC organization or contractor for life-cycle management. In such a situation, preparing the complete PDAD, LDD, and PDD after the application system has already been developed may not be useful. The As-Built System Documentation consolidates the requirements and design of an existing system in one product to facilitate the maintenance and future enhancement of the application system. This standard may *not* be used as a substitute for the PDAD, LDD, and PDD during the development of a new application system, because some material required for the successful design and development is not included in this standard.

The Technical Project Manager and members of the team responsible for system maintenance use this standard for documenting the requirements and design of a legacy system; key managers and quality assurance personnel use it for reviewing the as-built documentation; and the Executive Sponsor uses it when approving the document. The As-Built System Documentation is made available to all members of the maintenance or enhancement team, preferably in electronic form.

C. REFERENCE PUBLICATIONS

The following publications contain related information:

- SDLCM Methodology Handbook, Appendix E, Transition of Legacy Systems
- SDLCM Methodology Standard S–3051, Project Definition and Analysis Document
- SDLCM Methodology Procedure P–3111, Process Modeling
- 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–3164, External Systems Interface Diagrams

- SDLCM Methodology Standard S–3171, Logical Design Document
- SDLCM Methodology Standard S–3172, Physical Design Document
- SDLCM Methodology Standard S–3351, Data Dictionary
- *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

D. STANDARD

The As-Built System Documentation consolidates the requirements and design of an existing system in one product to facilitate the maintenance and future enhancement of the application system. A documentation product built to comply with this standard contains all of the requirements and design information essential for successful maintenance and future enhancement activities. As the application system is modified through maintenance or enhancement, the As-Built System Documentation will also be maintained to reflect the modified requirements and design.

The following paragraphs describe the content of the As-Built System Documentation. Tailor this standard to be consistent with the size scope, and complexity of the application system. When preparing this document product, retain all sections, including those that do not apply. Provide a brief statement to explain why a particular section is not applicable. Add other topics necessary to provide a complete picture of the system requirements and design. When appropriate, reference other documents rather than repeating material unnecessarily.

1. INTRODUCTION

1.1 Background

Describe the activities that led to the need to create this As-Built System Documentation.

1.2 Objectives

Describe the objectives of creating this documentation and briefly state how the document will be used to support maintenance and enhancement activities.

1.3 Scope

Describe what this product covers (its scope). Describe what it does not cover (constraints are related to scope and are to be addressed in this subsection).

1.4 Assumptions

State what assumptions have been made about the application system, its requirements, and its design. Discuss the risks, if any, associated with those assumptions.

1.5 Applicable Documents

List any other documents that apply.

1.6 Overview

Discuss the organization of the As-Built System Documentation. Describe the basic content of each major section of the document. Describe how this document will be maintained as the application system evolves.

PART 1—REQUIREMENTS

Part 1 of the As-Built System Documentation specifies the requirements of the existing application system. The requirements will be updated as the system evolves through maintenance and enhancement activities. This section is needed to ensure that the design and the system tests can be traced back to requirements.

2. SYSTEM REQUIREMENTS SPECIFICATION (SRS)

The purpose of this section is to document the system requirements specification (SRS) on which the design of the system was based. An SRS specifies the functional, performance, operational, and programmatic requirements for a system as an entity.

State individual requirements in quantitative terms. Avoid the use of negative requirements. Consider testability, completeness, consistency, and conciseness when writing each requirement.

Number all requirements uniquely to facilitate traceability.

2.1 Functional Requirements

Identify all system functional requirements based on the system and operations concept and other user needs or client requirements.

Organize the requirements into functional areas. For each functional area, define, as appropriate:

- Input data
- Process(es) (algorithms, transformations, manipulations, calculations)
- Output data
- External interfaces
- Communications
- Special management information needs

Use data requirements and system control as section headings, if appropriate.

Include a separate subsection for each functional area.

2.2 Performance Requirements

Decompose the performance requirements section into the following subsections, as appropriate:

- External workloads (for example, total size of data input per day)
- Internal function workloads (for example, function X must compute responses within 18 milliseconds)
- Throughput and response times (for example, complete simple query in the designated time period)
- Data quality, integrity, accuracy (for example, some elements must be accurate to a certain number of significant figures)
- Data retention (for example, number of years to retain data on-line versus long-term storage)
- System and communications capacity (for example, maximum number of communications lines connected to the system)
- Reliability, maintainability, and availability (for example, mean time between failures, mean time to repair)
- Periodicity, precision, simultaneity, frequency of transactions, processing, operations (for example, the system must process 10,000 transactions per second)
- Human workload and performance (for example, generation of a designated number of reports per day)
- Growth, flexibility, and expandability (for example, the system shall be capable of supporting four additional interfaces)
- Fault detection and isolation

2.3 Operational Requirements

Decompose the operational requirements section into the following subsections, as appropriate:

- Human factors (for example, user interface, keyboard layouts, display panel design, display formats).
- Environment (for example, temperature, humidity, air conditioning, shock and motion, shelf life)
- System monitoring
- Configuration control (for example, automated configuration control software)
- Training (for example, operator training)
- Support capabilities (for example, checkpoint and restart)
- Maintenance (for example, preventive maintenance)
- Logistics (for example, tracking of magnetic tapes)
- Facilities (for example, government-furnished equipment, test equipment, space, environmental such as heating and air conditioning)
- Safety (for example, fire, grounding of electrical equipment, requirements intended to prevent personnel injury)
- Security (for example, facility access, password protection, software, and data security requirements)
- Geographic location (for example, multiple sites)
- Documentation (for example, users or operations guides)

2.4 Programmatic Requirements

Decompose the programmatic requirements section into the following subsections, as appropriate:

- Maintenance facility
- Maintenance support requirements or constraints
- Special test requirements
- Installation, phase-in, and turnover
- Schedule and budget constraints
- Growth, flexibility, and expandability
- Maintenance standards
- Procurement activities
- Shipping and storage requirements

2.5 Special Requirements

Specify any system requirements that are not appropriate for inclusion in any of the previous sections.

2.5.1 ACCESSIBILITY REQUIREMENTS FOR INDIVIDUALS WITH DISABILITIES.

2.5.2 RECORDS MANAGEMENT REQUIREMENTS.

PART 2—LOGICAL DESIGN

Part 2 of the As-Built System Documentation specifies the logical design of the existing application system. The logical design will be updated as the system evolves through maintenance and enhancement activities.

3. APPROACH

3.1 Design Methods

Provide a high-level summary of the method(s) used to design 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 documented, analyzed, enhanced, or modified.

3.2 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.

3.3 Design Studies

If applicable and known, provide information on analysis, modeling, and trade studies performed during the original logical design effort or as a part of maintenance or enhancement redesigns. Show how the results of the studies (such as workload; performance; reliability, maintainability, and availability; and database management system) demonstrated that the logical design would meet requirements. List any assumptions used and present the results.

3.4 Design Issues

This section is applicable only when prior As-Built System Documentation is being updated for a maintenance change.

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.

4. 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 are identified in the Level 1 data flow diagram.

4.1 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 logical design. Include a context diagram. 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.

4.2 Logical Data Model

Provide a logical data model of the existing system, 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.

4.2.1 ENTITY RELATIONSHIP DIAGRAM

Include a copy of the system's entity relationship diagram (ERD) and the entity descriptions contained in the data dictionary. If the system depends on 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 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 document the approval to use the tool.

4.2.2 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 acts on (reads, manipulates, updates) these data entities.

Specify how the system navigates to legacy systems to obtain needed data. If the system uses special screens to access the legacy data, refer to the as-built screens or provide a first-cut design of the screens for a maintenance change.

Identify data entry and display requirements and provide the design for these items. Discuss screen and dialog performance requirements that are 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.

4.3 Logical Process Model

Discuss the process modeling activities, including the business processes and practices or decision-making processes that are automated and any transaction performance requirements, such as volume, speed, and frequency of transactions.

Copies of the system'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 5 of this document, so this section should refer the reader to Section 5 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.

5. SYSTEM DESCRIPTION

Include a copy of the Level 1 Data Flow Diagram (DFD) of the logical process model. For each process described in the Level 1 DFD, identify subsystems and configuration items (CIs), both those that are implemented in software (including COTS products) and those that are 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.

5.1 Subsystem Descriptions

5.1.1 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.

5.1.1.1 SOFTWARE Configuration Items

Provide a summary of the software CIs.

5.1.1.1.1 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 developed, purchased, or acquired from a combination of sources. List files and databases accessed by the CI.

5.1.1.1.*n* Software CI *n* Description

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

5.1.1.2 HARDWARE Configuration Items

Provide a summary of the hardware CIs.

5.1.1.2.1 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.

5.1.1.2.*n* Hardware CI *n* Description

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

5.1.*N* SUBSYSTEM *N* DESCRIPTION

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

5.2 External Interfaces

Provide a logical 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, include an external systems interface diagram (see standard S–3164).

If the logical description of the external interfaces will not be useful for system maintenance, this section is not applicable. In that case, provide only the physical description in Section 7.1.4.

5.3 Internal Interfaces

Provide a logical 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.

If the logical description of the internal interfaces will not be useful for system maintenance, this section is not applicable. In that case, provide only the physical description in Section 7.1.5.

6. SYSTEM OPERATIONS DESIGN

Provide an operational overview of the system.

6.1 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.

6.2 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.

PART 3—PHYSICAL DESIGN

7. PHYSICAL MODEL

7.1 System Architecture

7.1.1 SYSTEM COMPONENTS

Describe the physical characteristics of the network, client-server, or desktop environment on which the application system has been deployed. Include discussion of environmental constraints in terms of external communications, product compatibility or limitations, and any other technical issues.

For network architectures, characterize the network with respect to single- or multi-user capabilities and client-server architecture. Describe the network topology, connectivity, and protocols if they are significant considerations for the physical design. If applicable, describe physical security measures that are provided.

For client-server architectures, describe the physical characteristics of the file server system. Describe the different types of clients, and provide the configurations for each in terms of hardware and software. Identify any modifications or enhancements that were made.

Describe any additional servers such as those used for application and database services. For each additional server, describe its interface to the network and its security processes.

Describe printers, scanners, and other peripheral or physical components that are used with the application system.

7.1.2 SOFTWARE COMPONENTS

Describe the software components of the system and the functions that each provides.

Use structure charts or object-oriented diagrams expanded to the unit level, showing data flow, control, input, and output.

Describe the external interfaces of the software components; that is, the application interfaces for interprocess communication or data sharing that must be considered when replacing or upgrading a component.

For each COTS product, indicate if the product is used as-is to satisfy part or all of a requirement (or requirements), or describe how the product was customized or modified. Customization or modification methods include:

- Use configuration capabilities of the COTS product
- Use the Application Programming Interface (API) kits of the COTS products
- Create new functionality through the use of COTS programming tools
- Add and integrate additional COTS products

7.1.3 USER INTERFACE

Briefly describe how a user accesses and interfaces with the application system. For example, discuss the use of a common user interface, if applicable, to integrate desktop applications for word processing, spreadsheets, electronic mail, and database applications. Describe the types of security restrictions that are in place at the user interface.

Provide detailed descriptions of processing initiated by operator-specified input and actions in terms of control points, functions performed, and results obtained. Provide a listing of numbered error messages with descriptions of system and user actions.

Describe critical components of the interfaces such as the need for synchronization of operations (for example, data read and write) by either user data entry or automatic means.

7.1.4 EXTERNAL INTERFACES

Provide a description of each external interface (for example, interfaces to the network topology, another system, subsystem, or configuration item). 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. (See SDLCM Methodology Standard S–3164, External Systems Interface Diagrams.)

7.1.5 INTERNAL INTERFACES

Provide a description of each interface between and among the components of a configuration item. 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 format of the data components, including data element type, description, representation, range of values, units, etc.

7.2 Physical Data Model

7.2.1 GENERAL MAPPING OF THE LOGICAL MODEL OBJECTS TO THE PHYSICAL MODEL OBJECTS

Describe how the logical model was translated to the physical model. Describe any intermediate data structures that were generated and how they were generated. For example, if a tool was used, specify what tool and version of that tool was used.

For example, Table 4151–1 shows how the Logical Model Objects were translated to Physical Model Objects—the relational schema.

Table 4151–1. Logical to Physical Model Object Comparison

Logical Design Logical Model Objects	Physical Design Relational Schema
Entity Types	Relational Tables
Attribute Types	Columns
Relationship Types	Relational Schema Relationships
Information Types	Data Types
Subtype Set	Data Subtypes

If the general mapping of the logical to the physical model will not be useful for system maintenance, this subsection is not applicable.

7.2.2 SPECIFIC MAPPING OF THE LOGICAL MODEL OBJECTS TO THE RELATIONAL SCHEMA

List the relational schemas and their components, including:

- Relational tables that represent each entity
- Columns that represent attributes
- Relational schema relationships
- Database primary, foreign, and secondary keys

- Data types

Show how the relational schema components map back to the logical design using tables such as those shown in Tables 4151–2 through Table 4151–4.

Table 4151–2. Mapping of Entity and Attribute Types to Relational Tables and Columns

Logical Design Entity and Attribute Types	Physical Design Relational Tables and Columns
Entity Type 1	Relational Table 1
• Attribute Type A	• Column A, Primary Key
• Attribute Type B	• Column B
Entity Type 2	Relational Table 2
• Attribute Type C	• Column C, Primary Key
• Attribute Type D	• Column D
• Attribute Type E	• Column E
Entity Type *n*	Relational Table *n*

Table 4151–3. Mapping of Relationship Types to Relational Schema Relationships

Logical Design Relationship Types	Physical Design Relational Schema Relationships
Relationship Type 1	Relational Schema Relationship 1
Relationship Type 2	Relational Schema Relationship 2
Relationship Type *n*	Relational Schema Relationship *n*

Table 4151–4. Mapping of Information Types to Data Types

Logical Design Information Types and Subtype Sets	Physical Design Data Types and Data Subtypes
Information Type 1	Data Type 1
• Subtype Set A	• Data Subtype A
• Subtype Set B	• Data Subtype B
Information Type 2	Data Type 2
• Subtype Set C	• Data Subtype C
• Subtype Set D	• Data Subtype D
• Subtype Set E	• Data Subtype E
Information Type *n*	Data Type *n*

7.2.3 COMPLETING THE PHYSICAL DESIGN

Describe how the physical design was completed after the logical model was translated into the physical model. Include:

- Views specified
- Indexes identified
- Data structures that have been redefined for efficiency
- Column names that have been changed for clarity

7.2.4 DATA DEFINITION LANGUAGE

List the Data Definition Language (DDL) that was generated from the relational schema. Include data dictionary information, such as:

- Tables or view of tables and their definitions
- Columns for the tables or views showing:
  - ◊ Column names
  - ◊ The column's data type format and length
  - ◊ The column's definition
  - ◊ The column's data type set definition
  - ◊ The column's data type set comments

7.3 Physical Process Model

Document the process models that evolved from the logical design, including:

- Process Flow Diagrams
- Screen prototypes

7.3.1 PROCESS FLOW DIAGRAMS

Use process flow diagrams (See in SDLCM Methodology Procedure P–3111, Process Modeling) to document all automated processes. Expound on ideas from the logical model's data flow diagrams and process definitions.

7.3.2 SCREEN PROTOTYPES

Document the screen prototypes that evolved from the logical model's:

- Data flow diagrams
- Process Definitions
- Data flow view

For each screen, specify:

- Input fields

- Processing
- Output fields
- Dialog boxes
- Triggers

7.3.2.1 Screen Prototype 1

7.3.2.2 Screen Prototype 2

7.3.2.*N* Screen Prototype *N*

APPENDIX A. REQUIREMENTS-TO-DESIGN TRACEABILITY

Include a copy of the requirements-to-design traceability matrix, or refer to the on-line traceability matrix.

APPENDIX B. DATA DICTIONARY

Provide a copy of the system's data dictionary, or refer to the on-line data model and data dictionary.

APPENDICES. (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 As-Built System Documentation.

REFERENCES

List all cited references.