System Development and Life-Cycle Management (SDLCM) Methodology

Handbook, Version 2.3

July 2002

UNITED STATES NUCLEAR REGULATORY COMMISSION
WASHINGTON, DC

Foreword

Nuclear Regulatory Commission (NRC) Management Directive 2.5, “Application Systems Life-Cycle Management,” establishes the policies for applications systems life-cycle management. The System Development and Life-Cycle Management (SDLCM) Methodology implements Directive 2.5 by providing life-cycle structure and guidance to NRC Projects.

The SDLCM Methodology comprises seven components:

1. Define Initial Project Requirements
2. Acquire Support Resources
3. Design the Solution
4. Engineer the Solution
5. Deploy the Solution
6. Service the Solution
7. Decommission the Solution

The methodology is *not* Itself a document or a set of documents. It is *the approach to doing business* at NRC, and it is described by a set of documents, including but not limited to the following:

- *SDLCM Methodology Handbook*
- *SDLCM Methodology Procedures, Standards, and Forms*
- *SDLCM Methodology Tool Inventory*
- *SDLCM Methodology Overview Training*

Table of Contents

Figures

Tables

1. Introduction

1.1 Background

Within the Nuclear Regulatory Commission (NRC), every Project aims to provide its customer with an application system product that is engineered to satisfy the customer’s requirements, within determined cost, schedule, and quality guidelines. The ability to provide such a product is made easier when the Project team follows a comprehensive and consistent methodology. The System Development and Life-Cycle Management (SDLCM) Methodology provides life-cycle structure and guidance to NRC Projects.

1.2 Objectives

The objectives of this *SDLCM Methodology Handbook* are three-fold. First, this handbook defines the SDLCM Methodology application *system* life cycle and describes the component structure and each of the seven components. It also relates the component structure to the *software* development life cycle, which is embedded within two of the components.

Second, this handbook introduces some system and software engineering terminology and provides a common basis of understanding for all users of the methodology.

Finally, this handbook identifies each of the life-cycle roles that must be performed by management and technical personnel and clarifies the responsibilities of each of those roles.

1.3 Scope

This handbook discusses the SDLCM Methodology. It Includes the development and the Life-cycle management (That is, maintenance and enhancement) of NRC application systems from the definition of the initial Project requirements (After a Project has been identified) through the decommissioning of a system that is no longer to be used.

It does *not* include strategic technology and business systems planning.

Further, the SDLCM Methodology does not apply to the development and maintenance of NRC’s internal network and communications services infrastructure, which supports the applications systems.

1.4 Overview

Chapter 2 of this handbook presents an overview of the SDLCM Methodology. A careful reading of that chapter is recommended. The remainder of this document can be treated like a reference book. Chapter 3 provides guidance to a Project manager on selecting an appropriate software life-cycle model. Chapters 4 and 5 introduce the support activities of quality assurance (QA) and configuration management (CM), respectively. The next seven chapters describe in detail the seven components of the SDLCM Methodology.

Appendix A explains how to propose changes to the SDLCM Methodology or to the documentation set (including this handbook) that describes the methodology. Appendix B summarizes the roles introduced in this handbook and the responsibilities of the personnel who perform those roles. Appendix C discusses the required products of a Project that develops, enhances, or maintains a system using the SDLCM Methodology. Appendix D includes a summary list of all activities discussed in this handbook. Appendix E summarizes the process for transition of legacy systems from NRC to a contractor for life-cycle management. Appendix G is a glossary of important terms.

2. Methodology Overview

2.1 A Structured Approach

The SDLCM Methodology is a structured approach to designing, developing, deploying, maintaining, and decommissioning information systems. It addresses all aspects of an information systems solution from cradle to grave. It allows, and even encourages, flexibility within a clearly defined structure.

This handbook presents processes that clearly and unambiguously define what needs to be done, by whom, when, why, and how:

- Who? ⇒ Roles and their responsibilities
- What? ⇒ Activities and sub-activities
- When? ⇒ Sequence as illustrated in the diagrams
- Why? ⇒ Products
- How? ⇒ Tools and Techniques

Everyone involved in the application system development and maintenance process at NRC is required to use the SDLCM Methodology. Users include:

- Customers who are developing and maintaining their own systems with the involvement of the Office of the Chief Information Officer (OCIO)
- Contractors who are supporting customer- or OCIO-developed systems
- All OCIO personnel.

2.1.1 Seven Components

Figure 2–1 Illustrates the component structure of the SDLCM Methodology. Seven components are specified:

1. **Define Initial Project Requirements.**
   After the need for a Project has been established, identify the information management problem, clarify the scope, identify functional and data requirements, analyze alternative solutions, select appropriate tools, establish a Project plan, and develop a support resource request.

2. **Acquire Support Resources.**

   Obtain the necessary resources (for example, staff, technology, contractors, training) to ensure timely and effective progress on the Project.

3. **Design the Solution.**

   Analyze the (functional, data, communications, network, and interface) requirements, design the system solution, and prepare integration and Project plans.

4. **Engineer the Solution.**

   Acquire or construct all modules of the system, test the system, create the rollout strategy, and prepare the training materials.

5. **Deploy the Solution.**

   Install and roll out the integrated solution.

6. **Service the Solution.**

   Maintain or enhance an existing application system or its support environment.

7. **Decommission the Solution.**

   Inactivate and cease support of an application system.

Strategic technology and business systems planning activities (shown outside the broken line delineating the seven methodology components) are conducted prior to the start of Component 1. Those activities are outside the scope of the SDLCM Methodology and are not discussed in this handbook.

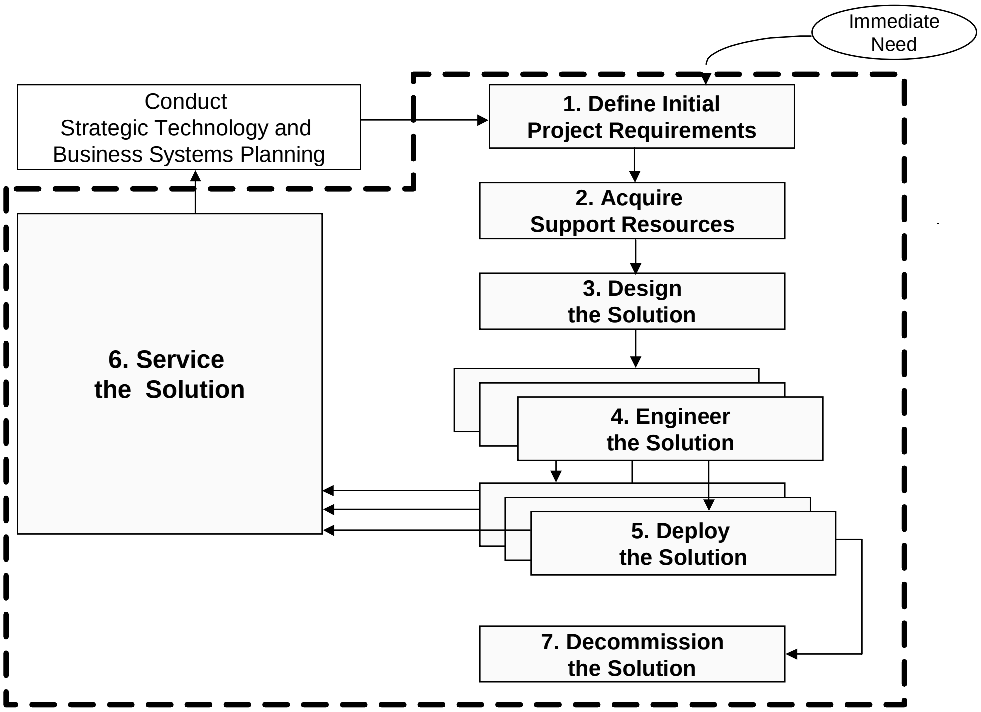

Immediate Need
Conduct Strategic Technology and Business Systems Planning
1. Define Initial Project Requirements
2. Acquire Support Resources
3. Design the Solution
4. Engineer the Solution
5. Deploy the Solution
6. Service the Solution
7. Decommission the Solution

Figure 2–1. Component Structure

2.1.2 Presentation of the Components

Subsequent chapters of this handbook describe each of the components in detail. The chapter for each component includes:

- A brief summary of the purpose of the component
- A table of roles and responsibilities
- Identification of the entry criteria, that is, the events that may trigger the activities within the component
- A list of inputs to the components

- A data flow diagram illustrating the overall flow of the activities performed within the component
- Identification of the outputs of the component
- A suggested list of techniques to support the performance of the activities and a pointer to a list of tools
- Identification of the exit criteria, that is, the conditions that are expected when the component is complete
- A more detailed description of each of the activities shown in the data flow diagram

2.2 Principles and Assumptions

The following *principles* are incorporated throughout the methodology:

- Strategic technology and business systems planning is conducted prior to the start of the Define Initial Project Requirement component of the methodology.
- When laws and mandates require specific activities to be completed, these activities are incorporated into the methodology at the appropriate time.
- All requests (new applications *and* upgrade or maintenance requests on existing applications) will be initiated through the first component, Define Initial Project Requirements.
- All systems requests will be evaluated for Business Process Reengineering (BPR) in the first component *before* proceeding to the other components within the methodology.
- OCIO will evaluate the end-user desktop environment and work with the user to resolve compatibility issues.
- The SDLCM Methodology (including both the process descriptions and the inventory of tools) will be reviewed and revised as required to support the evolving needs of NRC and also to implement improvements suggested by lessons learned from applying the methodology on NRC Projects.

The following *assumptions* are in place:

- Contractors May fill any of the Project roles *except* Executive Sponsor and Business Advocate.
- Systems analysts will be trained in facilitation, methodology, and specific tools as needed.
- The BPR techniques used will be standardized and will use automated tools that will be contained in the inventory.
- There will generally be a uniform environment with some variations permitted and managed as necessary.

2.3 Activities and Products

This section explains the terms used in Figure 2–2.

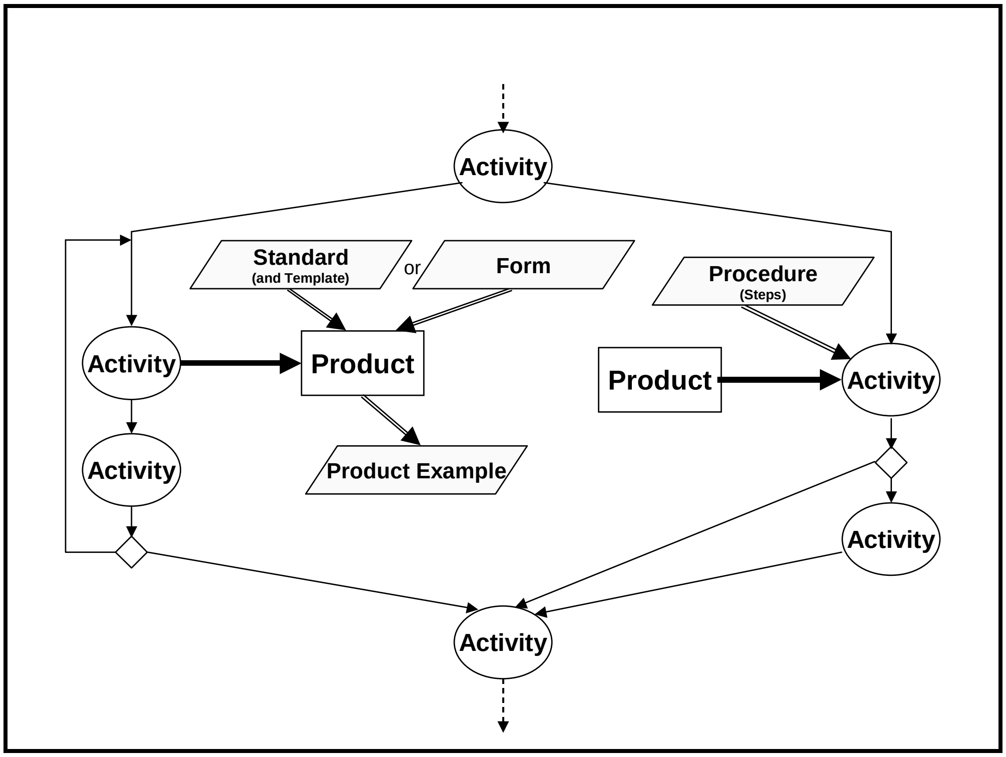


Figure 2–2. Relationships between Activities and Products

Each of the seven components of the methodology comprises a collection of *activities*. As suggested by the figure, 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.

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 this handbook or in a companion guidebook, then no separate procedure is provided

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. A product standard (for example, the standard for a Project Management 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 each product standard, 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 subsequent chapters and summarized in Appendix C. Product examples are maintained in the system documentation library. Procedures, standards, and forms are provided in a separate volume (see *SDLCM Methodology Procedures, Standards, and Forms*) that supports this handbook.

2.4 Feedback and Improvement

Every Project provides an opportunity to improve the process defined by the SDLCM Methodology. Figure 2–3 Illustrates the QA and process improvement feedback loops for Projects.

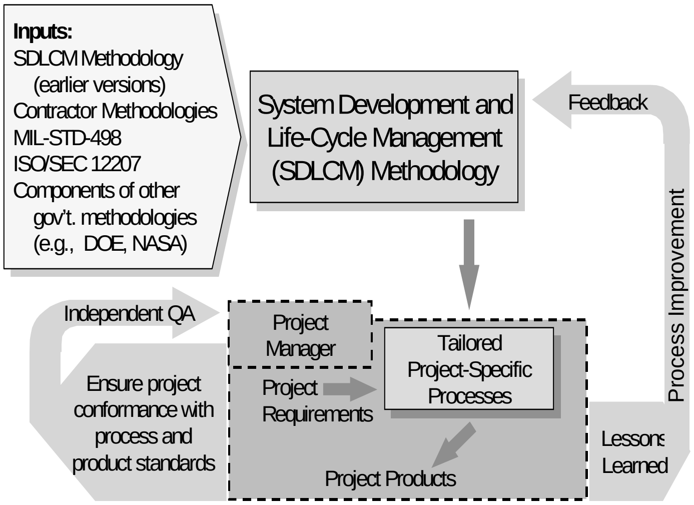

Inputs:
SDLCM Methodology (earlier versions)
Contractor Methodologies
MIL-STD-498
ISO/SEC 12207
Components of other gov't. methodologies (e.g., DOE, NASA)
System Development and Life-Cycle Management (SDLCM) Methodology
Feedback
Process Improvement
Independent QA
Project Manager
Tailored Project-Specific Processes
Project Requirements
Ensure project conformance with process and product standards
Project Products
Lessons Learned

Figure 2–3. A Living Methodology

The SDLCM methodology was originally based on the experiences of NRC and its contractors from earlier Projects and on understanding gained from studying other similar methodologies.

When a new Project is initiated, the Project manager reviews the methodology in the context of the Project requirements and derives a tailored Project-specific process, which is reviewed and approved by a QA representative. As the Project team applies the process to the development of products, the QA representative reviews the progress to ensure that both the process and the products conform to the approved standards. Any inconsistencies are reported to the Project manager. If the Project Manager fails to take appropriate action, the independent QA organization elevates the report to higher levels of management. The QA process is defined in more detail in Chapter 4.

Throughout the life of a Project, and especially as a part of Project closeout activities, lessons learned from applying the SDLCM Methodology to the Projects are fed back to the SDLCM Methodology Team for analysis. It is primarily the actual Project experience of managers and team members that sustains the process improvement feedback loop shown in the figure. The mechanism for reporting methodology deficiencies or suggesting improvements is described in Appendix A, Maintaining the SDLCM Methodology.

2.5 Systems Concepts

Figure 2–4 illustrates the concept of a system life cycle. This section clarifies some terms related to the system life cycle.

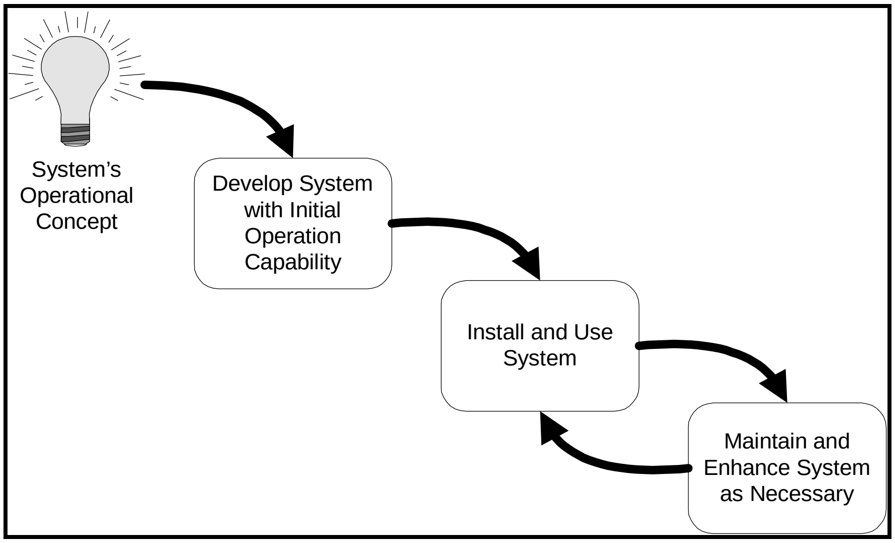

System's Operational Concept
Develop System with Initial Operation Capability
Install and Use System
Maintain and Enhance System as Necessary

Figure 2–4. Illustrative System Life Cycle

In this handbook, the term *system* (or *application system* or *information system*) refers to the operational entity that the NRC organization is responsible for developing, maintaining, or enhancing. That is, if the organization is responsible for developing several software and hardware configuration items (CIs) *and* is responsible for integrating them into an operational entity, then the collection of those CIs is the system. If, however, the organization is responsible for developing a single software CI, which may be integrated into (for example) a management information system by a different NRC organization, then the software CI itself is the system referred to in this handbook.

Some of the systems that NRC organizations develop, or maintain and enhance, include multiple CIs. There may be a mix of software CIs and hardware CIs, or the system may be only hardware or only software. This guidebook discusses the activities associated with the development and with the maintenance and enhancement of a system. When the system comprises only software CIs, then most of the system- and CI-level products are one and the same.

Development (see Figure 2–5) is the creation and installation of an operational system that meets an initially defined set of system requirements. Once the system is operational, subsequent changes are considered *maintenance* or *enhancement* (see Figure 2–6). Many of the maintenance and enhancement activities are the same or similar to those used in development. The SDLCM Methodology process applies equally to development and to maintenance and enhancement efforts.

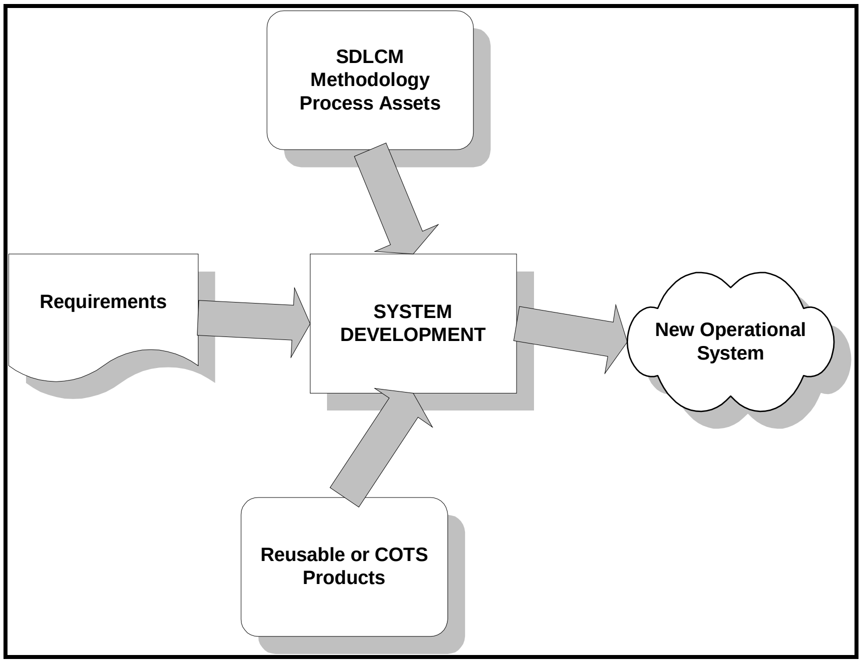


Figure 2–5. Development Context

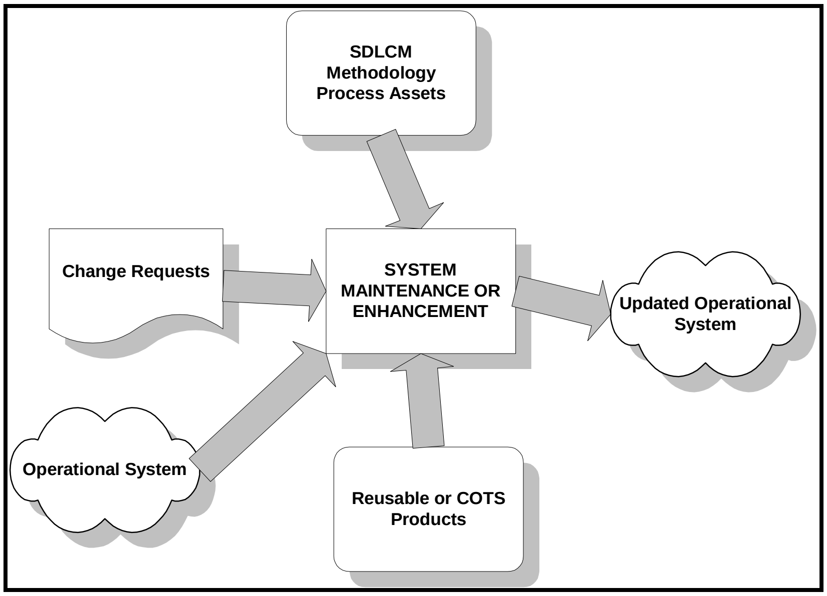

SDLCM Methodology Process Assets
Change Requests
SYSTEM MAINTENANCE OR ENHANCEMENT
Updated Operational System
Operational System
Reusable or COTS Products

Figure 2–6. Maintenance or Enhancement Context

3. Selecting an Appropriate Software Development Life-Cycle Model

The software development life cycle falls within the systems life cycle introduced in the preceding chapter. It also falls within Component 3 (Design the Solution) and Component 4 (Engineer the Solution) of the SCLCM Methodology. This chapter introduces four software development life-cycle models that may be applied to software development Projects at the NRC. Software enhancement Projects typically follow the same life-cycle model used for the original development activities. The maintenance life cycle (Component 6) is discussed in Chapter 3.2.

A life-cycle model comprises one or more phases (for example, a requirements definition phase, a design phase, a test phase). Each phase is defined as the time interval between two scheduled events. For example, in the waterfall life-cycle model, the design phase is defined as the period between the software specification review (SSR) and the critical design review (CDR).

Within each phase, one or more activities are executed. For example, during the waterfall model's design phase, the design activity is performed; the test planning activity may be done at the same time. In most cases, activities neither begin nor end precisely at the phase boundaries; rather, they overlap adjacent phases, as illustrated in Figure 3–1.

Notice that life-cycle phases are defined by time boundaries, whereas activities are defined by the type of work being performed.

Various methods (or techniques) may be used in the performance of an activity. For example, object-oriented design is one proven design method; structured design is another.

This handbook does not mandate any particular software life-cycle model, and the order of activities described here is not intended to conform to any particular model. Few specific methods are mandated for required activities. These decisions are left to the manager, who selects an appropriate life-cycle model and activity-related methods and defines them in the Project Management Plan. This chapter contains guidance on selecting an appropriate, recommended life-cycle model and methods for many activities.

For convenience, Table 3–1 provides the definitions (see the Glossary) of several important terms used extensively in this chapter.

Four life-cycle models are summarized in the following subsections:

- Waterfall development life-cycle model
- Incremental development life-cycle model
- Evolutionary development life-cycle model
- Package-based development life-cycle model

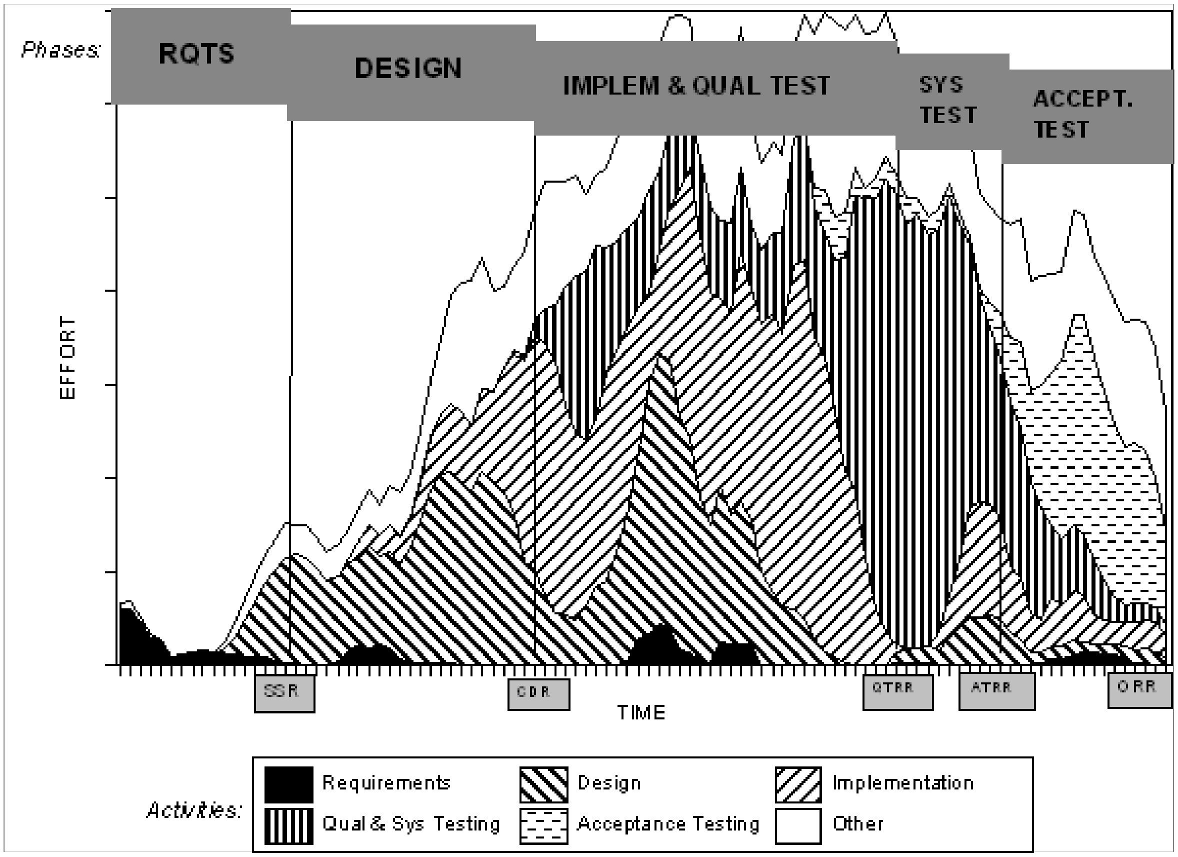


Figure 3–1. Phases and Activities

Table 3–1. Defining a Life Cycle

Term	Definition
Software life cycle	“The period of time that begins when a software product is conceived and ends when the software is no longer available for use.” [IEEE 610.12] a life cycle is typically divided into life-cycle phases.
Life-cycle model	A framework on which to map activities, methods, standards, procedures, tools, products, and services (for example, waterfall, and spiral).
Life-cycle phase	A division of the software effort into non-overlapping time periods. Life-cycle phases are important reference points for the software manager. Multiple activities may be performed in a life-cycle phase; an activity may span multiple phases.
Activity	A unit of work that has well-defined entry and exit criteria. Activities can usually be broken into discrete steps.
Method	A technique or approach, possibly supported by procedures and standards, that establishes a way of performing activities and arriving at a desired result.

3.1 Waterfall development Life-cycle Model

Table 3–2 Summarizes the Life Cycle Defined by the Waterfall development Model.

Table 3–2. Summary of Waterfall Development Life-cycle Model

Summary description and discussion	The waterfall (single-build) life-cycle model is essentially a once-through-do-each-step-once approach. Simplistically, determine user needs, define requirements, design the system, implement the system, test, fix, and deliver the system. This model is illustrated in Figure 3–2.
Advantages	• Well-studied, well-understood, and well-defined • Easy to model and understand • Easy to plan and monitor • Many management tools exist to support this life-cycle model
Disadvantages	• Most if not all requirements must be known up front • Does not readily accommodate requirements changes • Product is not available for initial use until the Project is nearly done
Most appropriate when ...	• Project is similar to one done successfully before • Requirements are quite stable and well-understood • The design and technology are proven and mature • Total Project duration is relatively short (less than a year) Customer does not need any interim releases

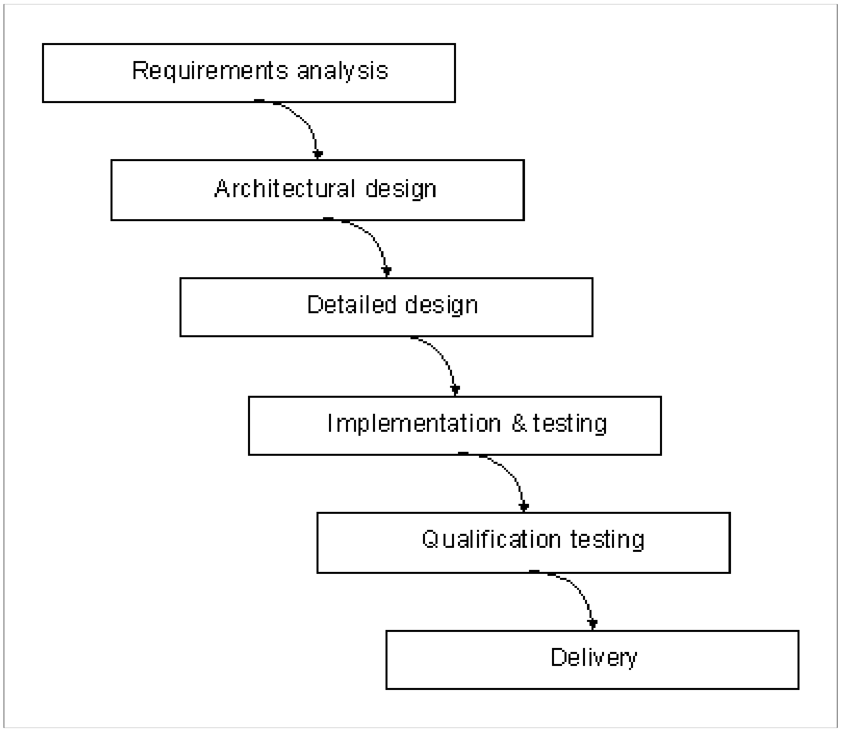


Figure 3–2. Waterfall Development Life-Cycle Model

3.2 Incremental Development Life-cycle Model

Table 3–3 summarizes the life cycle defined by the incremental development model.

Table 3–3. Summary of Incremental Development Life-Cycle Model

Summary description and discussion	The incremental (multi-build) life-cycle model determines user needs and defines a subset of the system requirements, then performs the rest of the development in a sequence of builds. The first build incorporates part of the planned capabilities; the next build adds more capabilities; and so on, until the system is complete. This model is illustrated in Figure 3–3.
Advantages	• Reduces risks of schedule slips, requirements changes, and acceptance problems • Increases manageability • Interim builds of the product facilitate feeding back changes in subsequent builds • Interim builds may be delivered before the final version is done; this allows end users to identify needed changes • Breaks up development for long lead time Projects • Allows users to validate the product as it is developed • Allows software team to defer development of less well understood requirements to later releases after issues have been resolved • Allows for early operational training on interim versions of the product • Allows for validation of operational procedures early • Includes well-defined checkpoints with customer and users via reviews
Disadvantages	• Like the waterfall life-cycle model, most if not all requirements must be known up front • Sensitive to how specific builds are selected • Places products (particularly requirements) under configuration control early in the life cycle, thereby requiring formal change control procedures that may increase overhead, particularly if requirements are unstable
Most appropriate when ...	• Project is similar to one done successfully before • Most of the requirements are stable and well-understood; but some TBDs may exist • The design and technology are proven and mature • Total Project duration is greater than one year or customer needs interim release(s)

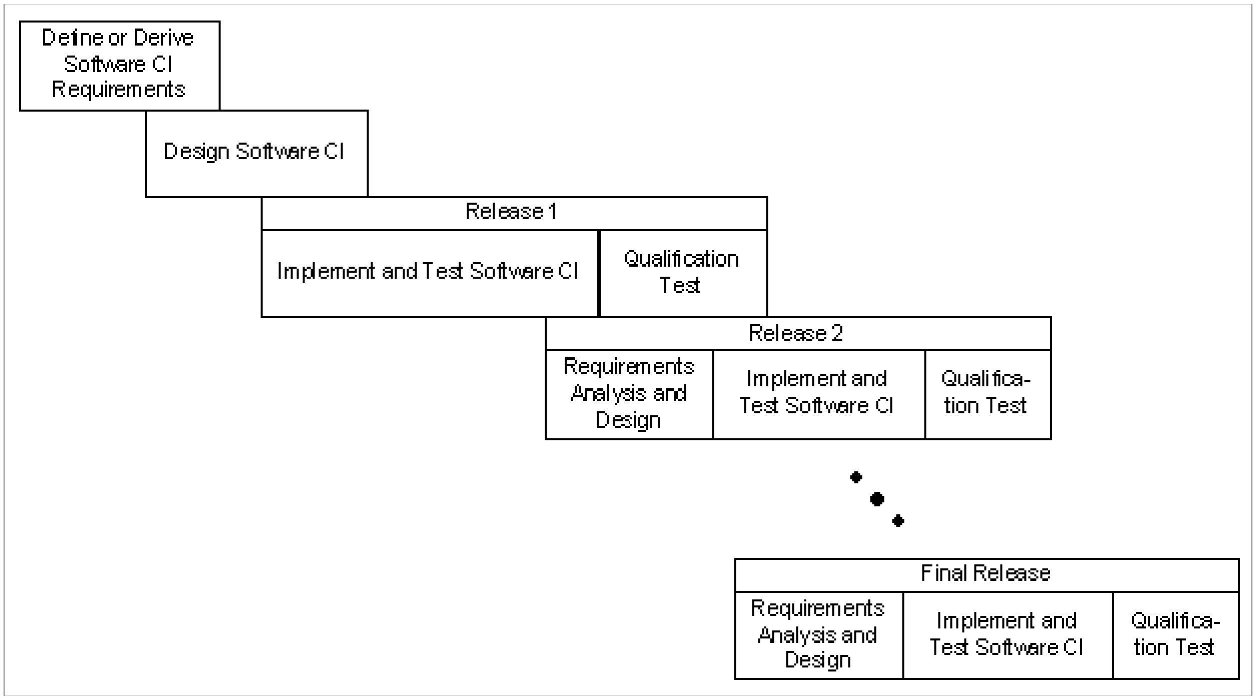

Define or Derive Software CI Requirements
Design Software CI
Release 1
Implement and Test Software CI
Qualification Test
Release 2
Requirements Analysis and Design
Implement and Test Software CI
Qualifica-tion Test
Final Release
Requirements Analysis and Design
Implement and Test Software CI
Qualifica-tion Test

Figure 3–3. Incremental Development Life-Cycle Model

3.3 Evolutionary Development Life-cycle Model

Table 3–4 Summarizes the Life cycle defined by the evolutionary development model.

Table 3–4. Summary of Evolutionary Development Life-Cycle Model

Summary description and discussion	Like the incremental development model, the evolutionary life-cycle model also develops a system in builds, but differs from the incremental model in acknowledging that the user needs are not fully understood and not all requirements can be defined up front. In the evolutionary approach, user needs and system requirements are partially defined up front, then are refined in each succeeding build. The system evolves as the understanding of user needs and the resolution of issues occurs. Prototyping is especially useful in this life-cycle model. (The evolutionary development life-cycle model is sometimes referred to as a spiral development model, but it is not the same as Boehm's spiral model. This model is also sometimes referred to as a prototyping life-cycle model, but it should not be confused with the prototyping technique. This life-cycle model is illustrated in Figure 3–4.

Advantages	• Not all requirements need be known up front • Addressing high risk issues (for example, new technologies or unclear requirements) early may reduce risk • Like the incremental life-cycle model, interim builds of the product facilitate feeding back changes in subsequent builds • Users are actively involved in definition and evaluation of the system • Prototyping techniques enable developers to demonstrate functionality to users with minimal of effort • Even if time or money runs out, some amount of operational capability is available
Disadvantages	• Because not all requirements are well understood up front, the total effort involved in the Project is difficult to estimate early. Therefore, expect accurate estimates only for the next cycle, not for the entire development effort. • Less experience on how to manage (progress is difficult to measure) • Risk of never-ending evolution (for example, continual “gold plating”) • May be difficult to manage when cost ceilings or fixed delivery dates are specified • Will not be successful without user involvement
Most appropriate when ...	• Requirements or design are not well-defined, not well-understood, or likely to undergo significant changes • New or unproved technologies are being introduced • system capabilities can be demonstrated for evaluation by users • There are diverse user groups with potentially conflicting needs

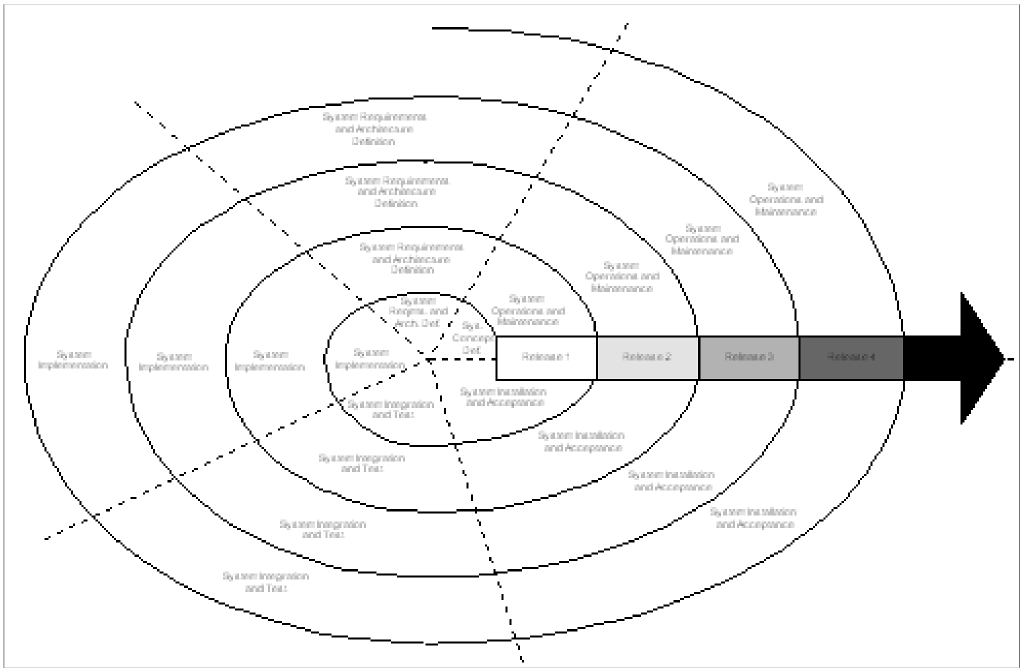

System Requirements and Architecture Definition
System Requirements and Architecture Definition
System Operations and Maintenance
System Operations and Maintenance
System Requirements and Architecture Definition
System Operations and Maintenance
System Rqmts. and Arch. Def.
System Operations and Maintenance
Sys. Concept Def.
System Implementation
System Implementation
System Implementation
System Implementation
Release 1
Release 2
Release 3
Release 4
System Installation and Acceptance
System Integration and Test
System Installation and Acceptance
System Integration and Test
System Installation and Acceptance
System Installation and Acceptance
System Integration and Test
System Integration and Test

Figure 3–4. Evolutionary Development Life-Cycle Model

3.4 Package-based Development Life-Cycle Model

Table 3–5 summarizes the life cycle defined by the package-based development model.

Table 3–5. Summary of Package-based Development Life-Cycle Model

Summary description and discussion	• The package-based development life-cycle model is used for system development based largely on the use of commercial-off-the-shelf and government off-the-shelf products and reusable packages. Typically, some custom software development is needed to provide interfaces among the non-developed items (NDIs). • This model is illustrated in Figure 3–5.
Advantages	• Lower cost than developing equivalent functionality from scratch • Cycle time also often lower than developing equivalent functionality from scratch • Improves confidence in quality of the end product (since quality of NDIs is already known)
Disadvantages	• May result in compromises between desired functionality and functionality provided by NDIs • Maintainability may be more of a challenge because source of NDIs may not be the same NRC organization (for example, requires third party to make changes, raises software CM issues when NDI vendor releases updated versions)
Most appropriate when ...	• A significant portion of the functionality of a system can be provided by NDIs

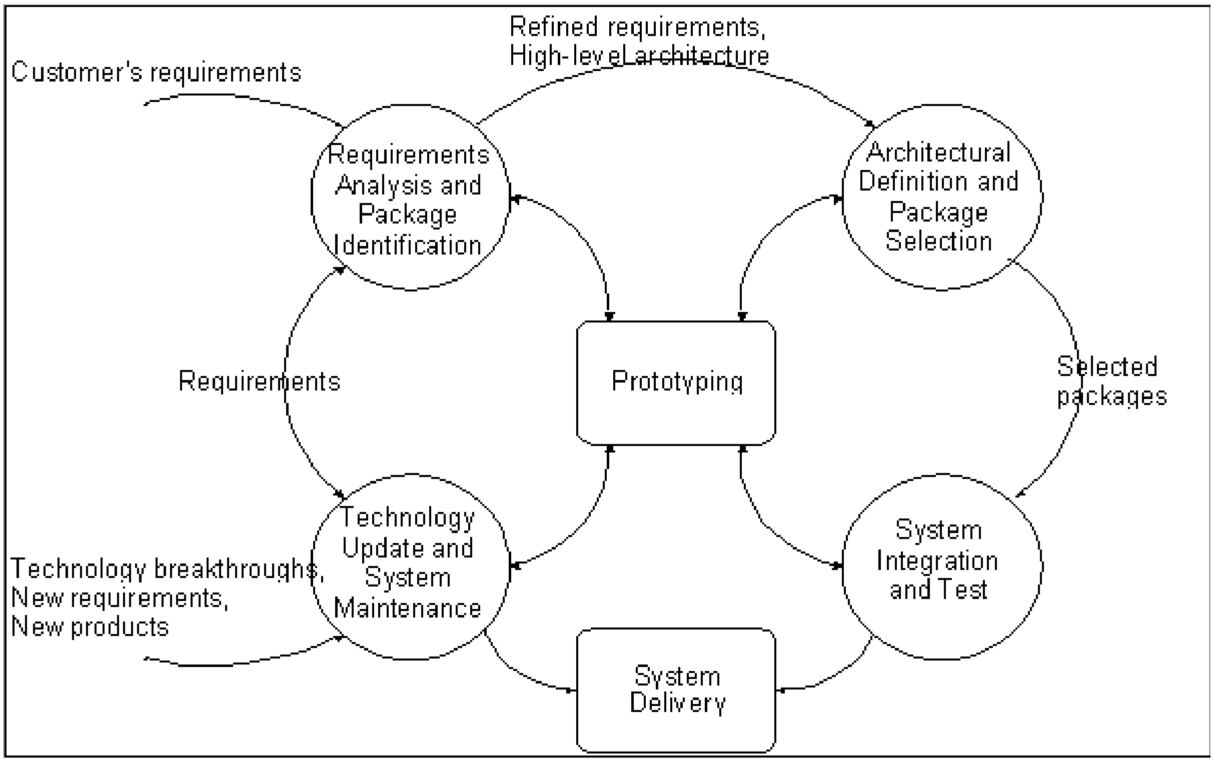

Refined requirements,
High-level architecture
Customer's requirements
Requirements Analysis and Package Identification
Architectural Definition and Package Selection
Requirements
Prototyping
Selected packages
Technology Update and System Maintenance
System Integration and Test
Technology breakthroughs,
New requirements,
New products
System Delivery

Figure 3–5. Package-Based Development Life-Cycle Model

4. Quality Assurance

4.1 Purpose

Quality Assurance (QA) is a planned and systematic set of activities whose purpose is to provide management with an independent view that approved processes are being used and that high quality products are being produced by NRC Project teams.

Quality Assurance involves:

- Reviewing and auditing the activities and products to verify that they comply with published SDLCM Methodology procedures and standards
- Providing managers and Project teams managers with the results of these reviews and audits

These review and audit activities occur throughout the life cycles of Projects and provide management with visibility needed to control the adherence to established plans, procedures, and standards.

4.2 Quality Assurance Goals

The Goals of NRC's Quality Assurance Program are to ensure that:

- The Quality Assurance activities, and the corresponding resource allocation, are planned.
- The products produced and activities performed at the NRC adhere to SDLCM Methodology procedures and standards.
- The results of any quality assurance activities are reported to the appropriate groups in a timely manner.
- Any noncompliance issues are corrected at the lowest possible level of management.

In pursuit of these goals at the Project level, Quality Assurance will:

- Support establishment of plans and procedures that meet contract and organization policies and adherence to the SDLCM Methodology procedures and standards.
- Participate and conduct reviews and audits. This includes Project activities and products.
- Identify compliance issues. Escalate as required.

4.3 Process Implementation

Each organization, within NRC or under contract to NRC, that initiates a Project for development or maintenance of systems that have an IT component is subject to the SDLCM Methodology, will prepare and implement a Quality Assurance Plan (QAP). The QAP describes the organization's tailored approach to QA. It identifies the group(s) responsible for performing QA and the relationships between QA and other parts of the organization (such as Program Management, Configuration Management, and the software development and maintenance teams). The plan identifies resources and includes a schedule for performing the required activities. Table 4–1 lists some important QA activities. The standard for preparation of the QMP is contained in S-2001.

4.4 Measurement and Analysis

Measuring the activities of the QA program permits management to evaluate the proficiency of the QA organization, which leads to better and more efficient planning of QA activities. Measurements may include but are not limited to:

- Completion of milestones for QA activities compared to a given Project schedule.
- Work completed, effort expended, and funds expended in the QA activities compared to the planned work.
- Number of process and product reviews and audits compared to the planned work.

4.5 Verifying Implementation

Organizational management holds periodic reviews of the QA program. These reviews are designed to provide management with awareness of and insight into current QA activities.

Table 4–1. QA Activities

QA Activities	QA Sub-activities
Process Assurance Cycle (PAC) Activities:	Document Project Approach
	Deviation/Waiver Review
	Pre-component Process Review
	Component Orientation Meeting
	In-progress Process Audit
	Life-cycle Phase Audit
Product Inspection and Certification	
Life-cycle Model Phase Review	
Document Review	
Audits:	Configuration Audit
	Document Pre-delivery Audit
	System Product Delivery Audit
	Inspection and Certification Meeting and Materials Audit
	Other In-process Audits
Non-conformance Reporting and Corrective Action	
Quality management Activities	
Independent Test Monitoring	
Training Monitoring	
Data Collection and Analysis Monitoring	
QA Activity Reports:	Weekly Reports to management
	Bi-weekly Reports to the Organizational Director of Quality
QA Calendar and Activity Schedules	
Input to Monthly Report	
PAC Related Reports	
Formal and Internal Review Report	Formal Review
	Formal Review Report
	Internal Review
	Internal Review Report
Audit Reports:	Life-cycle Phase Audit Report
	In-progress Audit Report
QA Records:	QA Task Files
	Task Area development Files
	Task Area Maintenance Files

5. Configuration Management

Configuration Management (CM) Is a discipline of applying administrative and technical procedures throughout the systems life cycle to:

- Identify, define, and baseline software and associated documentation in a system(s)
- Control modifications to and releases of the baseline
- Record and report the status of the baselines and modification requests
- Ensure baseline completeness, consistency and correctness
- Control storage, handling, and delivery

CM includes the following major activities:

- CM process implementation—definition and documentation of the configuration management activities
- Configuration identification—definition and identification of items subject to configuration control
- Configuration control—evaluation, coordination, and approval or disapproval of proposed changes to controlled items
- Configuration status accounting—recording and monitoring of changes to controlled items
- Data management—maintenance of official correspondence records, CM records, and controlled documentation
- Configuration evaluation—verification that controlled items meet their assigned requirements and are accurately documented

- In NRC Projects, configuration management is implemented at two levels, Application system, and Project/Task

5.1 CM Process Implementation

Each organization, within NRC or under contract to NRC, that initiates a Project for development or maintenance of systems that have an IT component and is subject to the SDLCM Methodology, is required to prepare and implement a Configuration Management Plan (CMP). The CMP describes the organization’s tailored approach to CM in conformance with the SDLCM Methodology. It identifies:

- The group(s) responsible for performing CM—such as the Configuration Management Office (CMO), Configuration Control Boards (CCBs), and Project CM representatives
- The relationships between CM and other parts of the organization—such as Project Management, Quality Assurance, and the software development and maintenance teams

The plan identifies resources and includes a schedule for performing the required activities. The standards for preparation of the CMP are contained in S-2501.

5.2 Configuration Identification

Configuration identification consists of selecting the configuration items (CIs) for a system and documenting their functional and physical characteristics. It provides the basis for applying management controls to a system configuration, permitting the isolation of items to be controlled, the tracking of their status, and the reporting of their configurations. The CMO, who is defined as the lead by the Overall Project Manager as the Technical Subject Matter Expert for CM activities, identifies those new software and hardware CI s for the Project. The CMO in cooperation with the Technical Project Manager establishes a scheme for the identification of software configuration items (SWCIs) and hardware configuration items (HWCIs) and their versions to be controlled for the Project.

For existing systems which are supported by legacy application systems, configuration identification begins with the documentation that established the baseline and its version, a complete inventory of items (including software) requiring conversion, rewrite, or transition, and associated hardware and communications (as required).

For new development, configuration identification begins with requirements that provide the work definition.

Configuration identification also applies to acquired hardware, communications equipment, Commercial-Off-The-Shelf (COTS) software, and documentation. A thorough configuration identification system is essential to monitoring, tracking, and implementing changes.

CI selection involves grouping system components into a unit subject to CM control. This activity is performed for software and hardware CI s by the systems engineering group lead by the Technical Project Manager, with approval from the appropriate Configuration Control Board (CCB). Non-software and hardware components (e.g., Management Directives, Rules, etc.) are the responsibility of the Project Level CCB with oversight by the System Executive Sponsor/Overall Project Manager. Certain functional, performance, and physical characteristics are allocated to each CI. A CI that is too large results in decreased visibility and ineffective control; one that is too small or placed under configuration control too early produces the opposite result: visibility that is at a very detailed level for the defined component and control that is excessive and therefore inefficient. For effective CM, CIs must be defined carefully and placed under configuration control at the appropriate time.

The following are guidelines by which CIs should be selected:

- Small and well-defined set of interfaces with other CIs
- Single source, such as a development group or contractor, responsible for providing the CI
- Largest entity that provides adequate client and management visibility into the development process
- Common schedule for acquisition, development, testing, and delivery of subordinate elements
- Independent operational or user interfaces from the user’s point of view
- Critical item with respect to safety, security, reliability, or other factors that require specific management attention
- As a single entity, requires maintenance, training, and logistical support
- Specific requirements for performance controls

- High degree of change or modification expected once the CI becomes operational

Because each CI requires its own baseline documentation and configuration control, identification of too many CIs must be avoided. If too many CIs are identified, system development will be excessively controlled and CM will be more complex and costly than necessary.

5.3 Configuration Control

Configuration control is the process of evaluating, approving or disapproving, and monitoring the implementation of changes to baselines during development, operation, or maintenance. This function includes the following:

1. Identification and recording of change proposals (SDLCM Methodology Form F–2502)
   - Approval or disapproval of the request
   - Implementation
   - Verification
   - Release of the modified software item
2. Establishing an audit trail that identifies
   - Each modification
   - Reason for the modification
   - Authorization of the modification

The CMO controls and audits all access to the controlled system items. Activities that handle safety or security critical functions requiring special handling are controlled in accordance with established NRC guidelines.

5.3.1 Responsibility and Process Flow

Configuration control is the responsibility of both the CMO and the CCB with oversight from the Executive Sponsor and Overall Project Manager.

The CMO is responsible for maintaining the integrity of all baselines. The CMO ensures that only NRC-approved changes are incorporated into baselines. CM-controlled baselines are updated using change control procedures that provide for systematic evaluation, coordination, and formal disposition of proposed changes and for implementation of changes that are approved.

The CCB is the heart of the configuration management process. The CCB provides a structured review process of requirements changes, problem or risk areas, and work in progress. The CCB is established under the authority and control of the NRC Executive Sponsor and functions in accordance with the CM program. The NRC Executive Sponsor, or a designated alternate, chairs the CCB, which is made up of technical managers, a QA representative, a CM representative (who serves as secretary), and other subject matter experts as required for a proper evaluation.

See SDLCM Methodology procedure P–2502, Change Proposal, for a detailed description of the change process and the steps that must be followed to propose and incorporate a change.

See SDLCM Methodology Procedure P–2501, Configuration Control Board, for a detailed description of the steps in the CCB process.

5.3.2Change Vehicles

Changes may be requested using several different vehicles as determined by the type of change. Note that some Projects may not require all of these mechanisms, and others may need additional vehicles. The following types of change requests are provided:

- Change Proposal (CP). A formal request to change an approved baseline (software, hardware, equipment, or their specification). See SDLCM methodology Procedure P–2502, Change Proposal, and Form F–2502, Change Proposal Form, for additional information.
- Request for Deviation or Waiver. A formal request to deviate from a standard methodology process or to be granted a waiver from following a standard process. In the context of CM the deviation or waiver typically requests a departure from established configuration baseline requirements. This type of request is discussed further in the chapter on Quality Assurance. See SDLCM Methodology procedure P–2001 deviations and waivers, for additional information.
- Problem Report (PR). A formal report of a problem detected during development, maintenance or operation of authorized systems. See SDLCM Methodology forms F–2251 (Problem Reports) and F–2252 (Problem Report Logs).

5.3.3Change Management

The management of proposed changes to a baseline is a critical part of system maintenance and, hence, of Component 6, Service the Solution, of the SDLCM Methodology. Therefore, the details of this element of configuration management are included in Section 11.2.

5.3.4 Release Management

The CMO formally controls the release and delivery of software products and documentation. Only formal agency guidance, and certified software and builds are formally delivered and implemented. Master copies of code and documentation are maintained for the life of the application system software product through the agency Central CM Library function. A prior release version and inactive application systems awaiting decommissioning are archived.

Planning and managing what changes go into each release are critical parts of system maintenance and, hence, of Component 6, service the solution, of the SDLCM Methodology. Therefore, the details of this element of configuration management are included in Section 11.3.

5.4 Configuration Status Accounting

Configuration status accounting is the recording, monitoring, and reporting of all changes to established baselines. The objectives of configuration status accounting are to ensure that:

- All approved changes are reflected in the affected baselines
- No baseline includes changes that have not been approved

The CMO generates management records and status reports that show the status and history of controlled items including software baselines. Status reports include the number of changes for a Project, latest versions, release identifiers, number of releases, and the differences between the releases.

5.5 Data Management

Data Management (DM) is closely related to configuration status accounting. While configuration status accounting focuses on recording and monitoring changes to controlled items, DM focuses on maintaining official correspondence records, CM records, and controlled documentation.

5.6 Configuration Evaluation

Configuration evaluation involves a series of inspections, tests, reviews, and audits to ensure that the manual processes and automated processes of the system, the data, and the guidance and automated documentation accurately represent the approved configuration baselines.

The functional completeness of the system's automated and manual process products (e.g., software code and policy/guidance) as compared to their defined requirements is determined using the Functional Configuration Audit (FCA). The FCA is the formal examination of functional characteristics of a configuration item, prior to acceptance, to verify that the item has achieved the requirements specified in its functional and allocated configuration documentation.

The physical completeness of the systems products (that is, whether all defined products were produced to the specifications called for by the required standards and guidance defined by the Project Management Plan and Software Development Plan) are determined using the Physical Configuration Audit (PCA). The PCA is a formal examination. The PCA for the as-built configuration of the automated components (application system code and executables) is the formal examination of the CI s against their technical documentation to establish and verify the CI's product baseline.

6. Component 1. Define Initial Project Requirements

6.1 Purpose

The purpose of this component is to:

- Identify an information management problem
- Clarify the scope
- Identify initial functional and data requirements
- Analyze alternative solutions
- Select appropriate tools
- Establish a Project plan
- Develop a support resource request
- Secure a Go decision for the Project

6.2 Roles and Responsibilities

The roles required to perform the activities in the define Initial Project Requirements component are shown in Table 6–1.

Table 6–1. Component 1 Roles and Responsibilities

Roles	Responsibilities
Business Advocate	Accountable for Execution
Executive Sponsor	Approval to Proceed
Overall Project Manager Business Project Manager Technical Project Manager Business Subject Matter Expert (SME) Technical SME Other Representatives or Stakeholders • Quality Assurance (QA) Manager • Configuration Management (CM) Manager • Records Management (RM)	Provides Input

6.3 Entry Criteria

Any of the following events may trigger the initiation of Component 1:

- Proactive business planning (Includes BPR)
- Proactive technology planning
- Identification of *ad hoc* business-driven requirements

- Identification of *ad hoc* technology-driven requirements

Note that these events occur as part of the “Conduct Strategic Technology and Business Systems Planning” box in Figure 2–1, and therefore are outside the scope of the SDLCM Methodology and this handbook. Any of these triggers may be driven by changes in:

- Cost
- Technology
- External mandate (for example, laws)

6.4 Input, Activities, and Outputs

The inputs to this component are as follows:

- Project Description (new and maintenance)
- Approved budget allocation
- Advance Procurement Plan
- Office or Regional approval
- Responsible Project contact
- Project responsibilities
- Technical Reference Model
- Enterprise Model
- Tool Inventory
- Any available Business Process Reengineering (BPR) documents

Figure 6–1 illustrates the flow of the following activities:

8. Clearly Identify Information Management Problem
9. Clarify Project Scope
10. Establish a Project Plan
11. Generate the Software Development Plan
12. Notify Records Management Branch to Review Records Management Requirements
13. Identify Requirements
14. Analyze Alternatives
15. Review Toolkit
16. Develop Support Resource Request
17. Plan for Deployment
18. Review (Component 1)

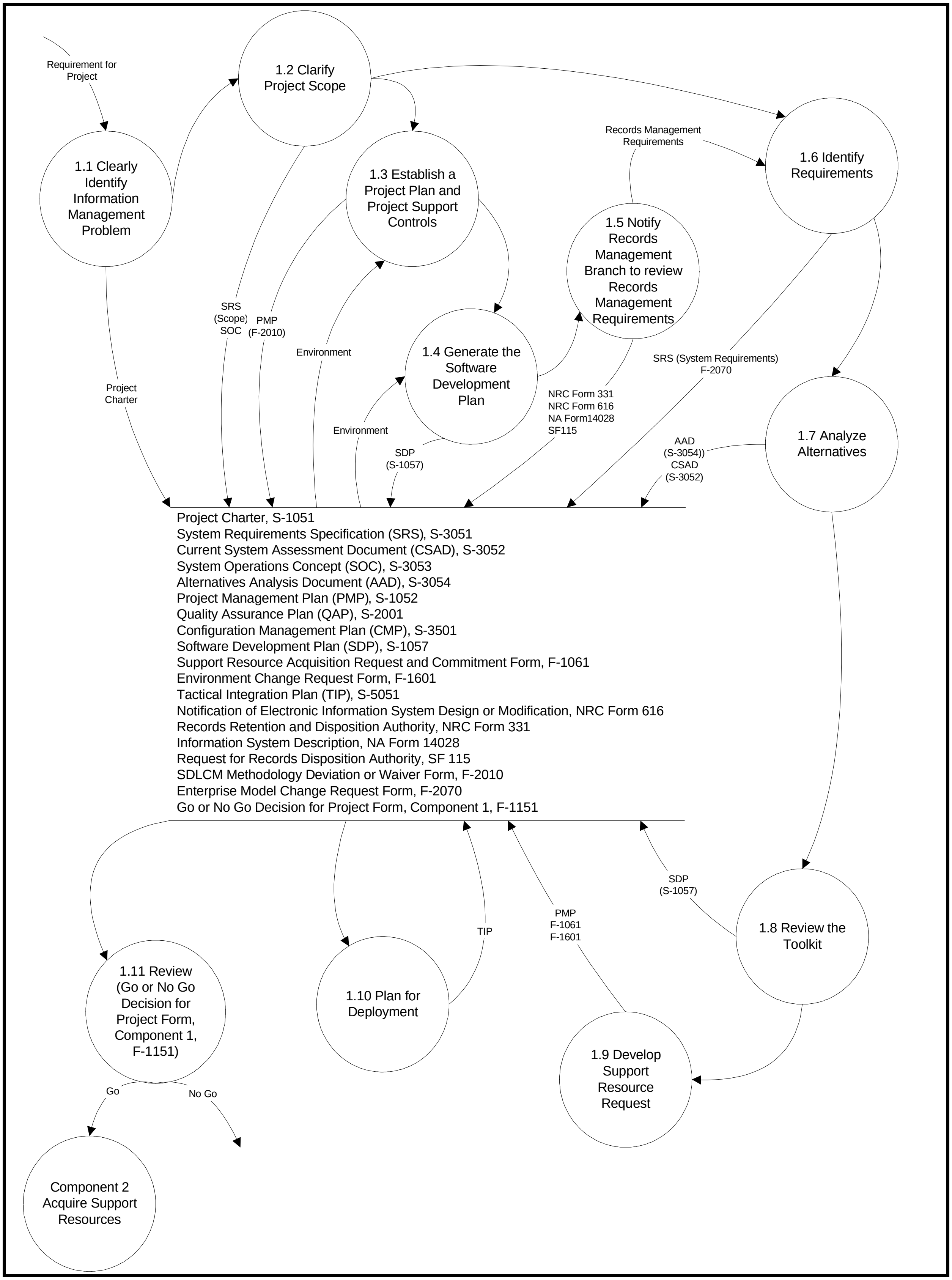

Requirement for Project
1.1 Clearly Identify Information Management Problem
1.2 Clarify Project Scope
1.3 Establish a Project Plan and Project Support Controls
1.4 Generate the Software Development Plan
1.5 Notify Records Management Branch to review Records Management Requirements
1.6 Identify Requirements
1.7 Analyze Alternatives
Records Management Requirements
Project Charter
SRS (Scope) SOC
PMP (F-2010)
Environment
Environment
SDP (S-1057)
NRC Form 331
NRC Form 616
NA Form14028
SF115
SRS (System Requirements) F-2070
AAD (S-3054))
CSAD (S-3052)
Project Charter, S-1051
System Requirements Specification (SRS), S-3051
Current System Assessment Document (CSAD), S-3052
System Operations Concept (SOC), S-3053
Alternatives Analysis Document (AAD), S-3054
Project Management Plan (PMP), S-1052
Quality Assurance Plan (QAP), S-2001
Configuration Management Plan (CMP), S-3501
Software Development Plan (SDP), S-1057
Support Resource Acquisition Request and Commitment Form, F-1061
Environment Change Request Form, F-1601
Tactical Integration Plan (TIP), S-5051
Notification of Electronic Information System Design or Modification, NRC Form 616
Records Retention and Disposition Authority, NRC Form 331
Information System Description, NA Form 14028
Request for Records Disposition Authority, SF 115
SDLCM Methodology Deviation or Waiver Form, F-2010
Enterprise Model Change Request Form, F-2070
Go or No Go Decision for Project Form, Component 1, F-1151
TIP
PMP
F-1061
F-1601
SDP (S-1057)
1.8 Review the Toolkit
1.9 Develop Support Resource Request
1.10 Plan for Deployment
1.11 Review (Go or No Go Decision for Project Form, Component 1, F-1151)
Go
No Go
Component 2 Acquire Support Resources

Figure 6–1. Component 1

As the figure suggests, many of the activities can be performed in parallel. For example, Activity 1.6, Establish a Project Management Plan, may be started as soon as there is enough information to begin planning; the activity cannot be completed, however, until all required planning information has been provided (for example, all system requirements must be known and all tool requirements must be determined).

The activities are described in more detail in Section 6.7.

Figure 6–1 also summarizes the outputs of all Component 1 activities in the central data store.

Appendix C includes a summary description of all products of NRC Projects, indicates within which components each product is created and updated, and specifies which products are required. The Current System Assessment Document (CSAD) is required to prioritize requirements even if there is no current automated system to be assessed. If there are no new requirements and only a relatively minor change is required, the prioritization can be documented adequately in the Software Engineering Notebook (SEN), which would be the case for a maintenance change. The SRS, Project Management Plan (PMP), and Tactical Integration Plan (TIP) are all begun as part of activities within this component and are all updated in subsequent components.

The products, whose standard and form numbers are shown in the figure, are described in more detail in the companion volume *SDLCM Methodology Procedures, Standards, and Forms*.

6.5 Techniques and Tools

The activities of Component 1 are supported by the following techniques.

- Business Area Analysis
- Business Process Reengineering
- Data Modeling
- Process Modeling
- Structured Walkthrough
- Peer Review

Refer to the current *SDLCM Methodology Tool Inventory* for the recommended set of tools.

6.6 Exit Criteria

Component 1 is complete when:

- All major products have been reviewed (structured walkthrough or peer review)
- All major products have been inspected by QA and certified to conform with Project standards
- All products have been placed under Project-Level CM control
- Information has been shared
- Issues have been resolved
- There is management buy-in to move forward
- There are no reasons to stop this Project now

6.7 Component 1 Activity Details

The following pages provide detailed activities of Component 1, Define Initial Project Requirements.

Activity 1.1: Clearly Identify Information Management Problem

See Figure 6–2. Using the Outputs from the Strategic Technology and Business systems Planning Activities (Which Are Outside the Scope of this Methodology):

1. Determine What Business Processes and Organizations Are Within the Scope of this Project.
2. Determine the Business Goals, Objectives, or Problems to Be Achieved.
3. Summarize the Findings in the Project Charter.

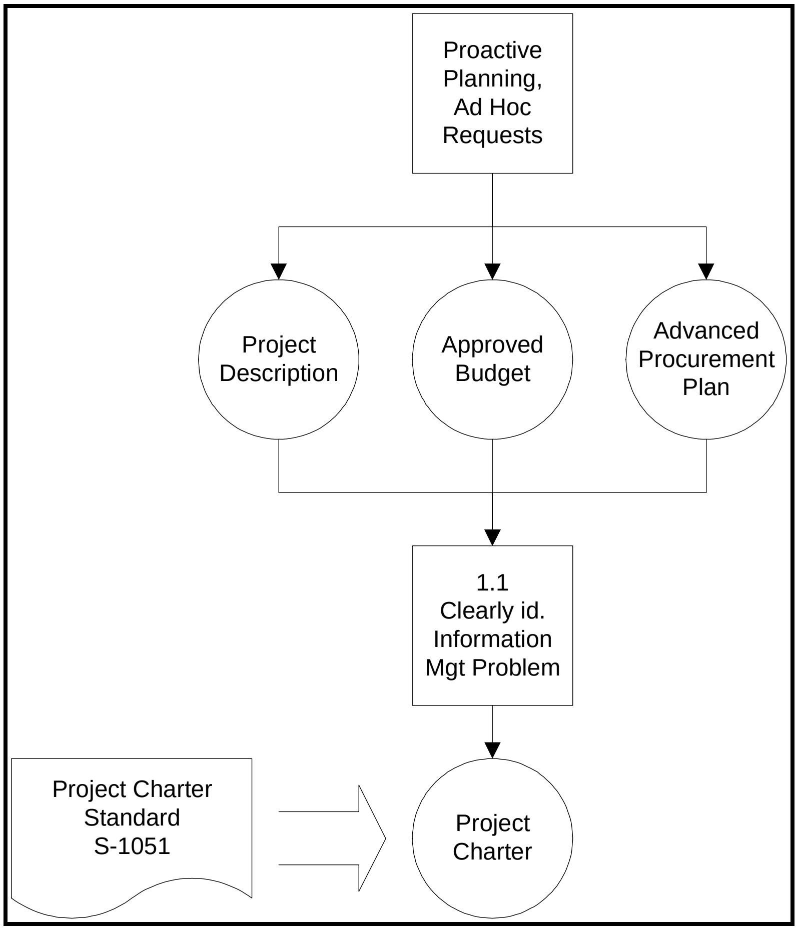

Proactive Planning, Ad Hoc Requests
Project Description
Approved Budget
Advanced Procurement Plan
1.1 Clearly id. Information Mgt Problem
Project Charter Standard S-1051
Project Charter

Figure 6–2. Identify Information Management Problem

Activity 1.2: Clarify Project Scope

See Figure 6–3. Using the Project Charter developed in Activity 1.1, and the existing Technical Reference Manual and Enterprise Model:

1. Clarify the scope of the Project.
2. Document this in the Scope section of the SRS.
3. Generate the SOC.

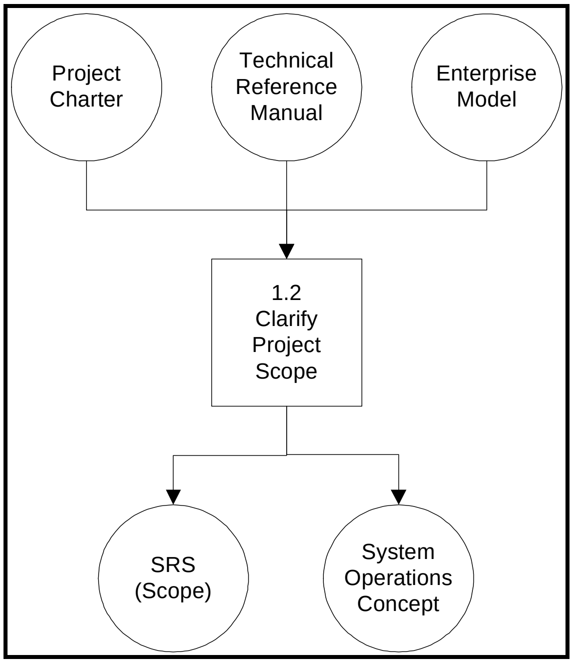

Project Charter
Technical Reference Manual
Enterprise Model
1.2 Clarify Project Scope
SRS (Scope)
System Operations Concept

Figure 6–3. Clarify Project Scope

Activity 1.3: Establish a Project Plan and Project Support Controls (CM, QA)

See Figure 6–4. Using the Project Charter, the SRS, and the System Operations Concepts as Input:

1. Develop a Project Management plan. On the basis of the complexity of the Project, choose an appropriate life-cycle model to develop the Project. For example, if this is a very straightforward enhancement, a waterfall approach may be appropriate. If the requirements are not well defined, then an iterative development approach with active involvement of the customer may be more appropriate.
2. Use any tools and methods available to estimate the amount of effort required to develop the needed functionality.
3. On the basis of the amount of effort required and the complexity of the Project, develop a tactical plan. That is, define a Project team with the right mix of senior and junior people, and define their roles and responsibilities.
4. Define intermediate milestones and products.
5. Define Project critical success measures, including performance requirements.
6. Evaluate and plan for records management.
7. Specify CM requirements and package results in the CMP.
8. Specify QA requirements and package results in the QAP.
9. Package the results of your work in the Project Management Plan.
10. Have QA review your plan and make any necessary changes.
11. Have Project-Level CM capture your plan in the Project-Level CM library.

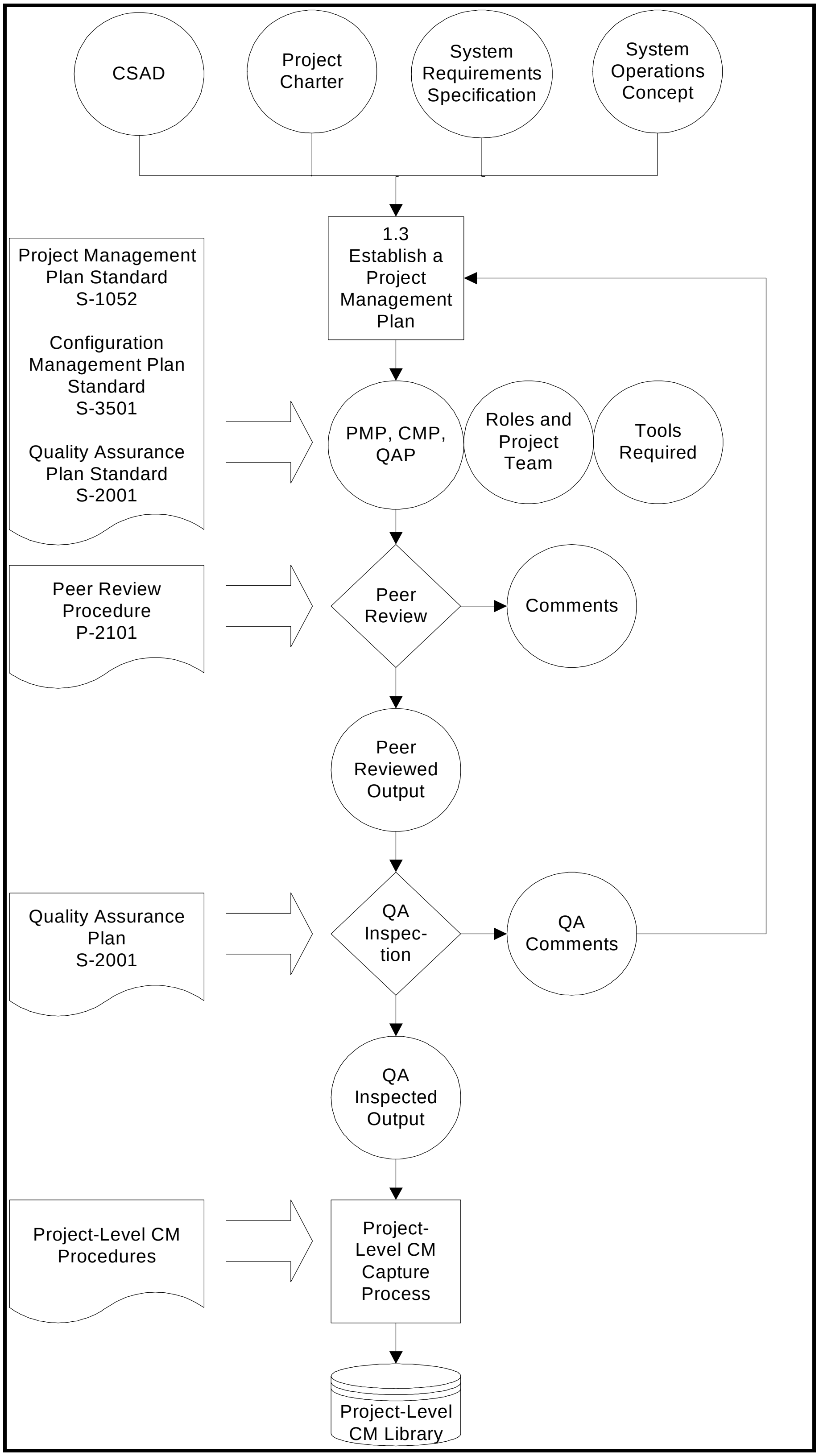

CSAD
Project Charter
System Requirements Specification
System Operations Concept
1.3 Establish a Project Management Plan
Project Management Plan Standard S-1052
Configuration Management Plan Standard S-3501
Quality Assurance Plan Standard S-2001
PMP, CMP, QAP
Roles and Project Team
Tools Required
Peer Review Procedure P-2101
Peer Review
Comments
Peer Reviewed Output
Quality Assurance Plan S-2001
QA Inspec- tion
QA Comments
QA Inspected Output
Project-Level CM Procedures
Project- Level CM Capture Process
Project-Level CM Library

Figure 6–4. Establish a Project Plan

Activity 1.4: Generate the Software Development Plan

See Figure 6–5. Using the Project Management Plan developed in Activity 1.3, the existing Technical Reference Model, schedule, and information influencing appropriate life-cycle, generate the Software Development Plan.

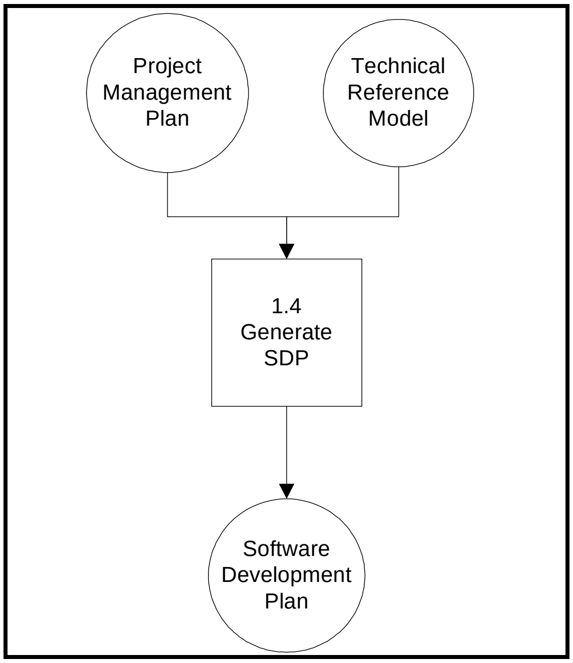


Figure 6–5. Generate the Software Development Plan

Activity 1.5: Notify Records Management Branch to Review Records Management Requirements

See Figure 6–6. Using information from the Project Charter and the scope portion of the SRS:

1. Complete the indicated government forms and deliver them to the Records Management Branch to notify them that an application system is being newly developed or enhanced.
2. After the forms have been processed by records management personnel, establish contact with the designated representative of the records management organization.
3. Add the legal and regulatory records management requirements to the user requirements documented in the SRS in Activity 1.2

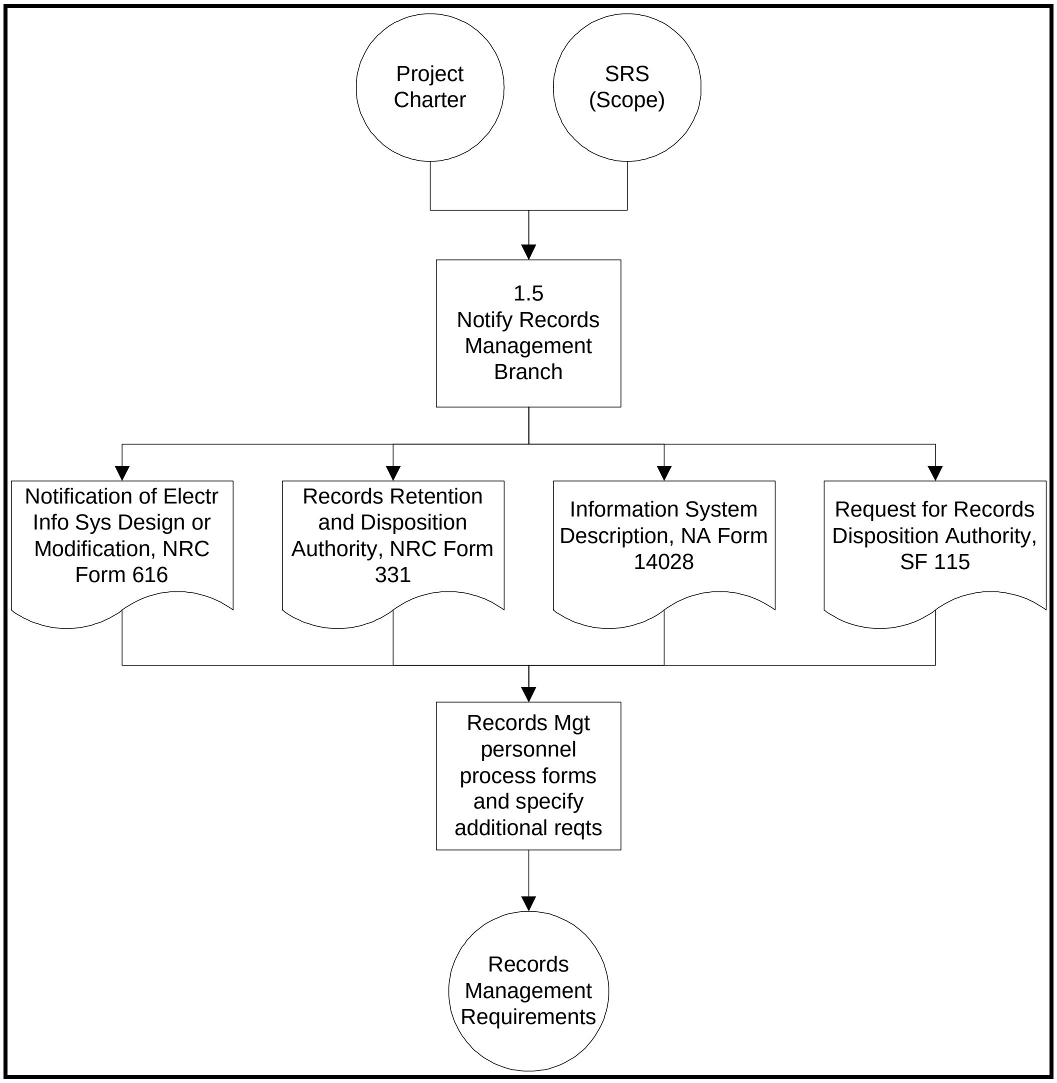

Project Charter
SRS (Scope)
1.5 Notify Records Management Branch
Notification of Electr Info Sys Design or Modification, NRC Form 616
Records Retention and Disposition Authority, NRC Form 331
Information System Description, NA Form 14028
Request for Records Disposition Authority, SF 115
Records Mgt personnel process forms and specify additional reqts
Records Management Requirements

Figure 6–6. Notify Records Management

Activity 1.6: Identify Requirements

See Figure 6–7. Using the scope section of the SRS, any additional input from Records Management, and the existing Technical Reference Manual and Enterprise Model:

1. Review the architecture in the Enterprise Model.
2. Document the relationships between the proposed Project and the information system architecture in the Enterprise Model.
3. Review the current technology in the Technical Reference Model.
4. Assess the impact of the Project on current technology architecture in the Technical Reference Model.
5. Develop a Project context diagram to show how this Project fits into the existing Technical Reference Model and Enterprise Model.
6. Develop an initial entity list.
7. Develop preliminary entity definitions for the entities listed above.
8. Evaluate legal requirements.
9. Evaluate policy/guidance requirements.
10. Evaluate public accessibility.
11. Identify physical requirements.
12. Identify data requirements.
13. Identify functional requirements.
14. Identify security requirements.
15. Identify human factors including user business knowledge requirements.
16. Package the results of your analysis in the System Requirements Specification (SRS), mapping the components of your analysis to the sections of the SRS as defined in the standard.
17. Have QA review the SRS to ensure that it meets requirements. Make corrections as needed.
18. Have Project-Level CM function capture the updated SRS in the Project-Level CM Library.

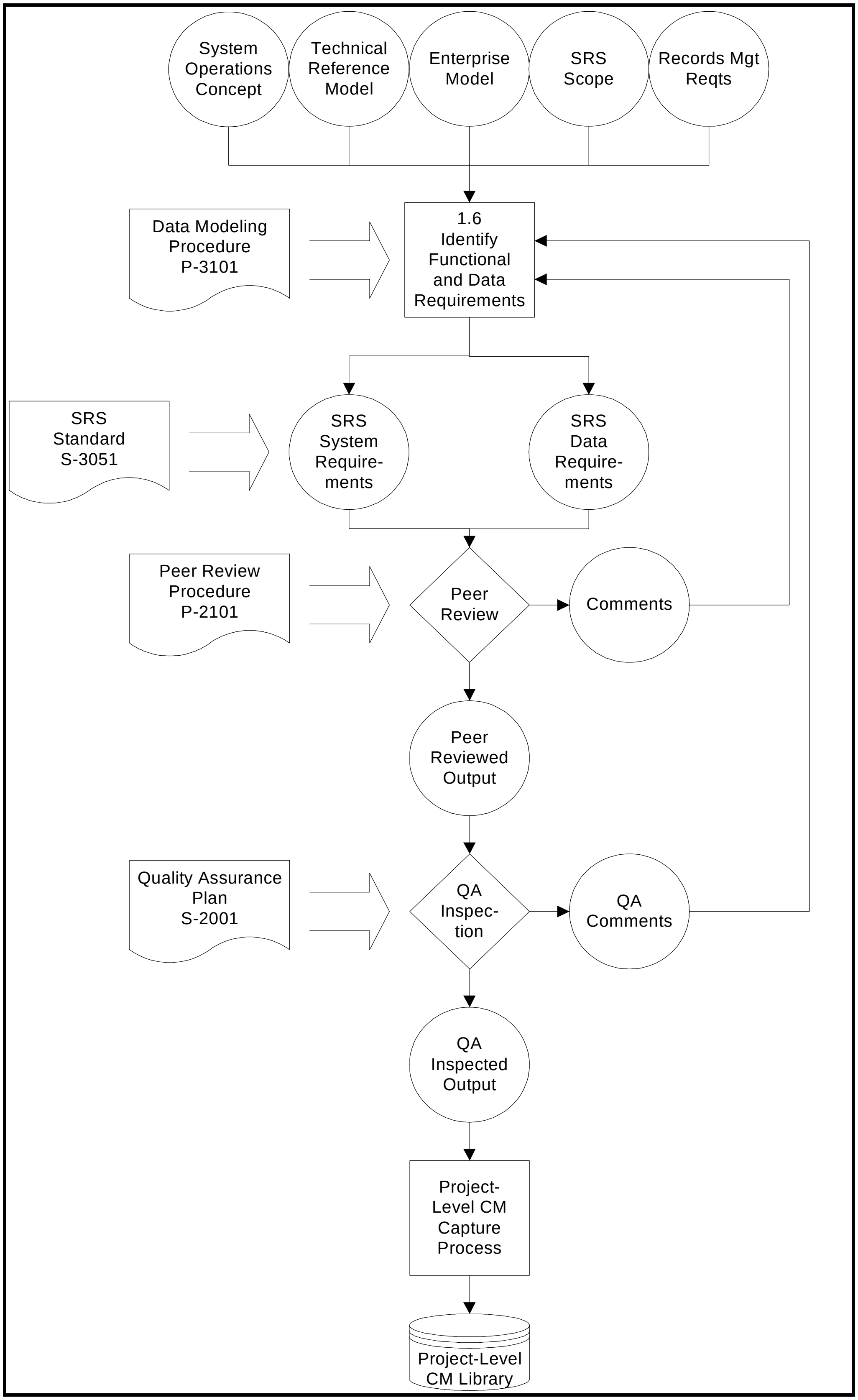

System Operations Concept
Technical Reference Model
Enterprise Model
SRS Scope
Records Mgt Reqts
Data Modeling Procedure P-3101
1.6 Identify Functional and Data Requirements
SRS Standard S-3051
SRS System Require-ments
SRS Data Require-ments
Peer Review Procedure P-2101
Peer Review
Comments
Peer Reviewed Output
Quality Assurance Plan S-2001
QA Inspec-tion
QA Comments
QA Inspected Output
Project-Level CM Capture Process
Project-Level CM Library

Figure 6–7. Identify Functional and Data Requirements

Activity 1.7: Analyze Alternatives

See Figure 6–8. This activity includes the assessment of current systems, the analysis of alternative approaches for satisfying the requirements, and the development of an operational concept.

Using the requirements documented in the SRS, determine whether it is better to enhance, supplement, or replace any existing systems. Use a process such as the following:

1. Create a table for assessing the current system(s). Use the sample shown in the CSAD standard.
2. Review the list of requirements in the SRS. Enter these functions in the requirements column of the table.
3. Identify the existing systems that might be used to help solve the information management problem. For each existing system, determine which requirements are satisfied and enter those in the appropriate column of the table. Then note the remaining requirements satisfied by none of the current systems in the final column of the table.
4. Document the results of the assessment in the CSAD. If there is no current system, the CSAD is still required to the level of completion of Section 4.
5. Validate the requirements defined in the SRS by comparing them to the System Operations Concept (S-3052), created in activity 1.2. If additional requirements are discerned, update the SRS, as appropriate.
6. Analyze the alternative approaches for satisfying the remaining requirements. Consider the advantages and disadvantages of enhancing, supplementing, or replacing each current system that was assessed and options for developing a new system if nothing is reusable. This analysis continues with the examination of using GOTS or COTS, or developing new to satisfy the prioritized requirements. The examination should weight and record sufficient information to determine the best options for satisfying the requirements and document the costs based on a five year design/operation time frame, the risks and the benefits of each alternative (see Appendix B, Management Directive 2.2, Capital Planning and Investment Control). Document the results in the Alternatives Analysis Document (S-3054).
7. Based on Alternatives Analysis, update System Operations Concept (S-3052), as appropriate, to reflect best approach resulting from the analysis.
8. Have QA review the Assessment of Current System, Analysis of Alternatives, and System Operations Concept and recommend any necessary corrections.
9. Have Project-Level CM function capture these documents in the Project-Level CM Library.

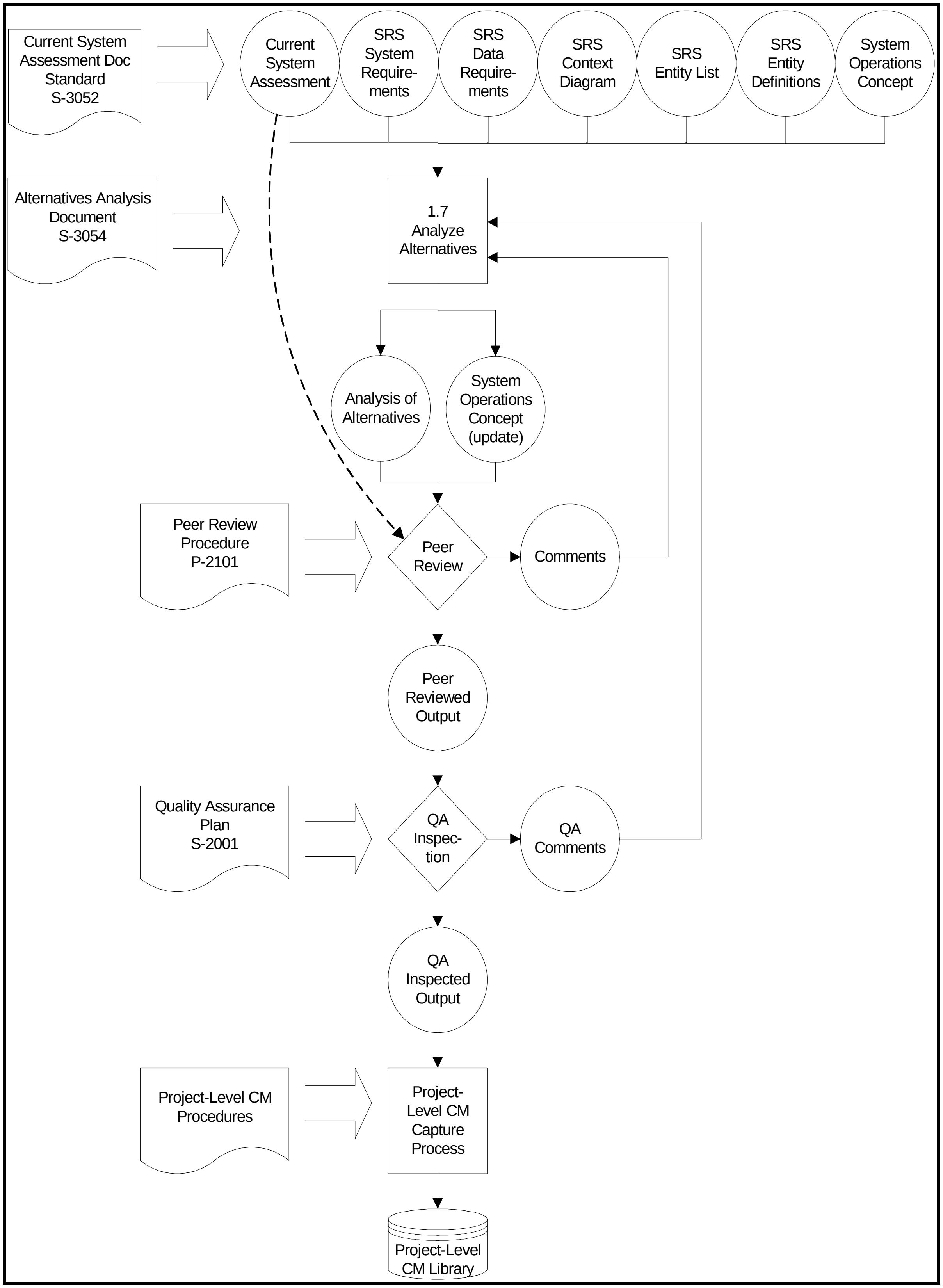

Current System Assessment Doc Standard S-3052
Current System Assessment
SRS System Require-ments
SRS Data Require-ments
SRS Context Diagram
SRS Entity List
SRS Entity Definitions
System Operations Concept
Alternatives Analysis Document S-3054
1.7 Analyze Alternatives
Analysis of Alternatives
System Operations Concept (update)
Peer Review Procedure P-2101
Peer Review
Comments
Peer Reviewed Output
Quality Assurance Plan S-2001
QA Inspec-tion
QA Comments
QA Inspected Output
Project-Level CM Procedures
Project-Level CM Capture Process
Project-Level CM Library

Figure 6–8. Analyze Alternatives

Activity 1.8: Review the Toolkit

See Figure 6–9. Using the results of your Analysis of Alternatives and the System Operations Concept:

- Review the Tool Inventory to select preferred tools or identify new tool requirements. This information will be used to identify your resources request in Activity 1.9.
- Update SDP, as needed.

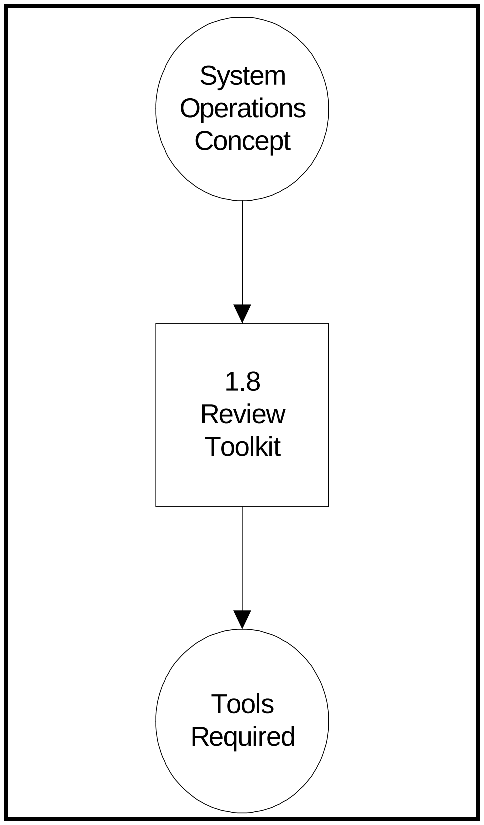


Figure 6–9. Review the Toolkit

Activity 1.9: Develop Support Resource Request

See Figure 6–10. Using the results of your detailed planning (as documented in the Project Management Plan), which shows the effort, people, computer hardware and software requirements, and any methods or tools necessary, request the resources you will need to be successful.

1. Update PMP schedules based on SDP updates.
2. Use an Environment Change Request Form and the process defined in the Environment Change Procedure to request any new tools or technologies.
3. Use a Support Resource Acquisition Request and Commitment Form (F-1061), If Needed, to request personnel and any other resources.
4. Identify procurement needs in detail.
5. Negotiate to obtain a commitment for the resources you need.

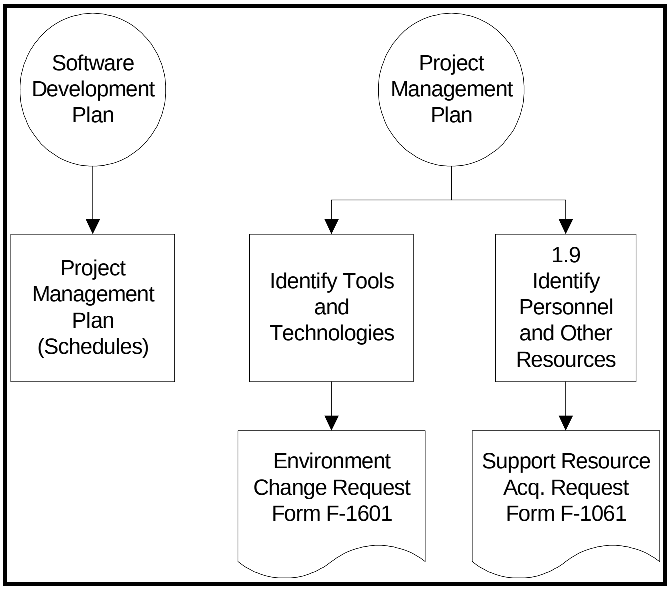

Software Development Plan
Project Management Plan
Project Management Plan (Schedules)
Identify Tools and Technologies
1.9 Identify Personnel and Other Resources
Environment Change Request Form F-1601
Support Resource Acq. Request Form F-1061

Figure 6–10. Develop Support Resource Request

Activity 1.10: Plan for Deployment

See Figure 6–11. Using all information provided by the Project Team and all information gathered to support the development of the Project Charter, SRS, SDP, and PMP, begin to plan for deployment by developing the initial version of the TIP. Document everything that is currently known about the deployment. Where necessary, indicate that sections of the TIP will be updated or completed as activities of subsequent components.

As suggested by Figure 6–1, development of the TIP may be started concurrently with development of the other major products of Component 1

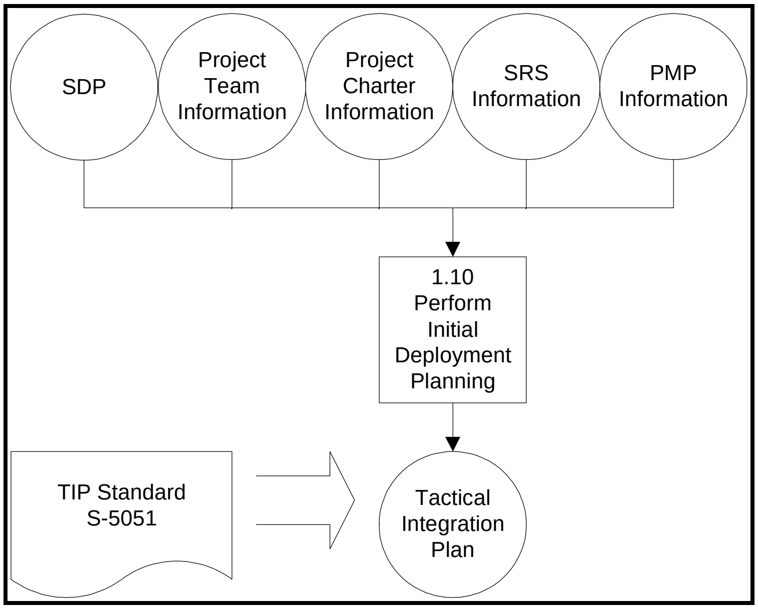

SDP
Project Team Information
Project Charter Information
SRS Information
PMP Information
1.10 Perform Initial Deployment Planning
TIP Standard S-5051
Tactical Integration Plan

Figure 6–11. Plan for Deployment

Activity 1.11: Review (Component 1)

See Figure 6–12. Review the Project to make a GO or NO-GO Decision.

1. Justify the Project. Review the need for the Project, the cost in terms of personnel and resources, and the return on investment for the Project. Consider the best case scenario and the worst case scenario of moving forward with the Project.
2. Complete the form required obtaining approval to proceed.

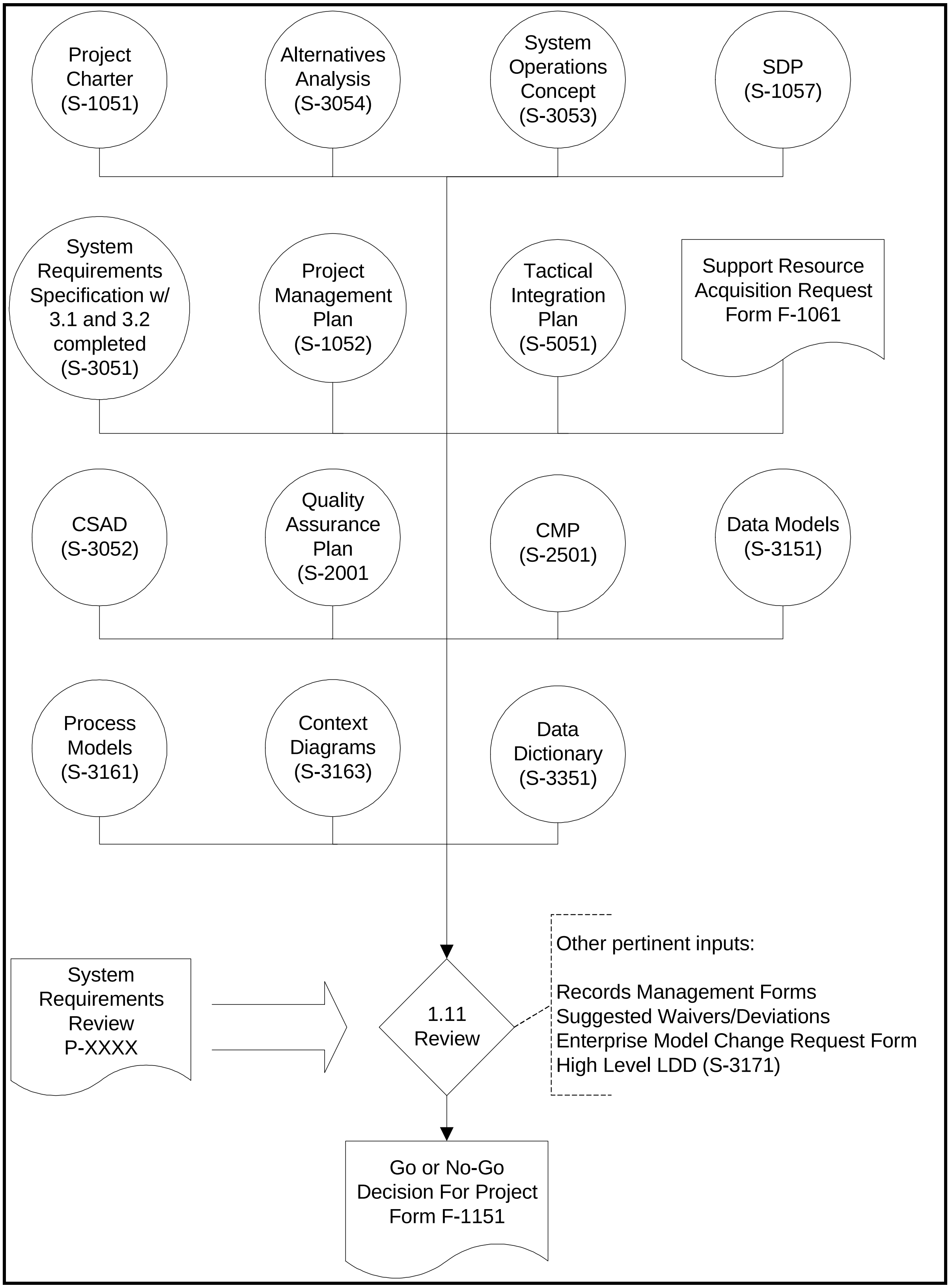


Figure 6–12. Review Component 1

7. Component 2. Acquire Support Resources

7.1 Purpose

The purpose of this Component is to obtain the necessary resources to ensure timely and effective progress on the Project, including:

- Staff
- Technology
- Contractors
- Training

7.1.1 Projects and Funding Vehicles

It is very important to understand that NRC's SDLCM Methodology deals with Projects and products, *not* with contracts and deliverables. This section of the chapter on Component 2, Acquire Support Resources, relates the Project-oriented methodology to the contracting process. The remainder of the documentation set of the SDLCM Methodology is free of contract-dependent terminology, so it can be applied on any contract vehicle.

NRC's typical mechanism for providing a source of Project funding to a contractor is the Contract Line Item (CLIN).

The three terms *CLIN/Delivery Order*, *Project*, and *system* have three distinct meanings:

- A CLIN/Delivery Order is a vehicle for defining discreet portions of the work to be performed under contract and for applying funding for a Project, part of a Project, or multiple Projects.
- a *Project* is an undertaking requiring an organized effort, focused on developing or maintaining a specific product. Typically, a Project has its own funding, cost accounting, and product schedule. That is, the Project is the work to be done and the personnel assigned to perform the work; it is not the same as the product (for example, a *system*) that the Project personnel are to produce.
- a *system* is an operational entity. An application system or information system supports a business requirement.

A Project produces *Products*. A CLIN/Delivery Order requires the delivery of *Deliverables*.

NRC designed the SDLCM methodology to be applied to a Project rather than to a CLIN/Delivery Order.

7.1.2 Funding Approaches and Project Products

A Project may be funded all at once under a single CLIN or Delivery Order, incrementally under a single CLIN or Delivery Order, or incrementally under a sequence of CLINs.

The document products specified by the SDLCM Methodology are to be developed as products of a complete Project, not necessarily as deliverables of a single CLIN/Delivery Order. If a CLIN/Delivery Order funds only a portion of a Project, it then stands to reason that not all Project products will be delivered under that CLIN/Delivery Order. In any case, the Overall Project Manger is responsible for ensuring that the Project products (as specified by the SDLCM Methodology) are produced (or formally waived) whether or not they are contractually specified as CLIN/Delivery Order deliverables. The Project Management Plan, for example, is a key Project management tool, which must be updated whenever any change results from the issuance of a new CLIN/Delivery Order or the modification to an existing CLIN/Delivery Order.

The ideal contracting approach (see Figure 7–1) is to fund a single Project under a single CLIN/Delivery Order (perhaps with incremental additions of funds). For example, a possible approach is to fund Component 1, so that the contractor can help with the definition of the requirements, and then to fund and execute Component 2 to modify the funding vehicle (not to issue a different one) to acquire support for Components 3, 4 and 5 as a unit. Separating 3 and 4 and funding them separately is not reasonable because together they comprise the software development life cycle portion of the SDLCM Methodology (see Chapter 3). Depending on the life-cycle model chosen (again see Chapter 3 for examples of life-cycle models), it may be logically impossible to separate Components 3 and 4. If the same contractor supports the system deployment in Component 5, or if the deployment is phased along with the design and engineering aspects, then it is also not feasible to separate Component 5 and all three components must be treated as a unit under a single CLIN/Delivery Order.

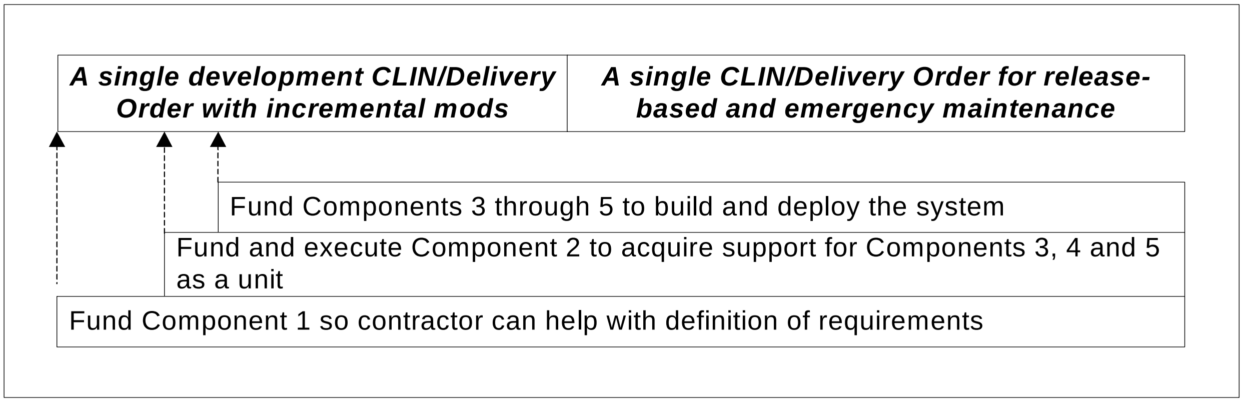

A single development CLIN/Delivery Order with incremental mods
A single CLIN/Delivery Order for release-based and emergency maintenance
Fund Components 3 through 5 to build and deploy the system
Fund and execute Component 2 to acquire support for Components 3, 4 and 5 as a unit
Fund Component 1 so contractor can help with definition of requirements

Figure 7–1. Ideal Project Funding Approach

7.2 Roles and Responsibilities

The roles required to perform the activities in the Acquire Support Resources component are shown in Table 7-1.

7.3 Entry Criteria

The following of the following may trigger the initiation of Component 2:

- Approved Project from the Define Initial Project Requirements component (Component 1)
- Change of priorities
- Change in commitments of staff

Table 7–1. Component 2 - Roles and Responsibilities

Roles	Responsibilities
Overall Project Manager Business Project Manager Technical Project Manager	Accountable for Execution
Executive Sponsor	Approval to Proceed
Business Advocate Development Team Business SMEM Technical SME Other Representatives or Stakeholders • Quality Assurance Manager • Configuration Management Manager	Provides Input

7.4 Input, Activities, and Outputs

The inputs to this component are as follows:

- Project Management Plan
- System Requirements Specification
- Tactical Integration Plan
- Support Resource Acquisition Request and Commitment
- New Tool Requirement
- Staffing (current and planned)
- Advance Procurement Plan (outside the scope of this methodology)
- Budgets (outside the scope of this methodology)

Figure 7–2 illustrates the flow of the following activities:

1. Specify the work to be done
2. Staff the Project
3. Train the staff
4. Acquire and install other required resources
5. Update the Project Management Plan

6. Continue Deployment Planning
7. Review (Component 2)

The inputs to this component were produced as products of Component 1 or prior to the start of the Project and are in the Project-Level CM library.

As the figure suggests, many of the activities can be performed in parallel. The activities are described in more detail in Section 7.7.

Figure 7–2 also summarizes the outputs of all Component 2 activities in the central data store.

Appendix C includes a summary description of all products of NRC Projects, indicates within which components each product is created and updated, and specifies which products are required.

The products, whose standard and form numbers are shown in the figure, are described in more detail in the companion volume *SDLCM Methodology Procedures, Standards, and Forms*.

7.5 Techniques and Tools

The activities of Component 2 are supported by the following techniques.

- Structured Walkthrough
- Peer Review

Refer to the current *Technical Reference Model* for the recommended set of tools.

7.6 Exit Criteria

Component 2 is complete when:

- All critical products have been reviewed (structured walkthrough or peer review)
- Key products have been inspected by QA to conform to Project standards
- Products have been placed under CM control
- Information has been shared
- Issues have been resolved
- There is management buy-in to move forward
- There are no reasons to stop this Project now

7.7 Component 2 Activity Details

The following pages provide detailed activities of Component 2, Acquire Support Resources.

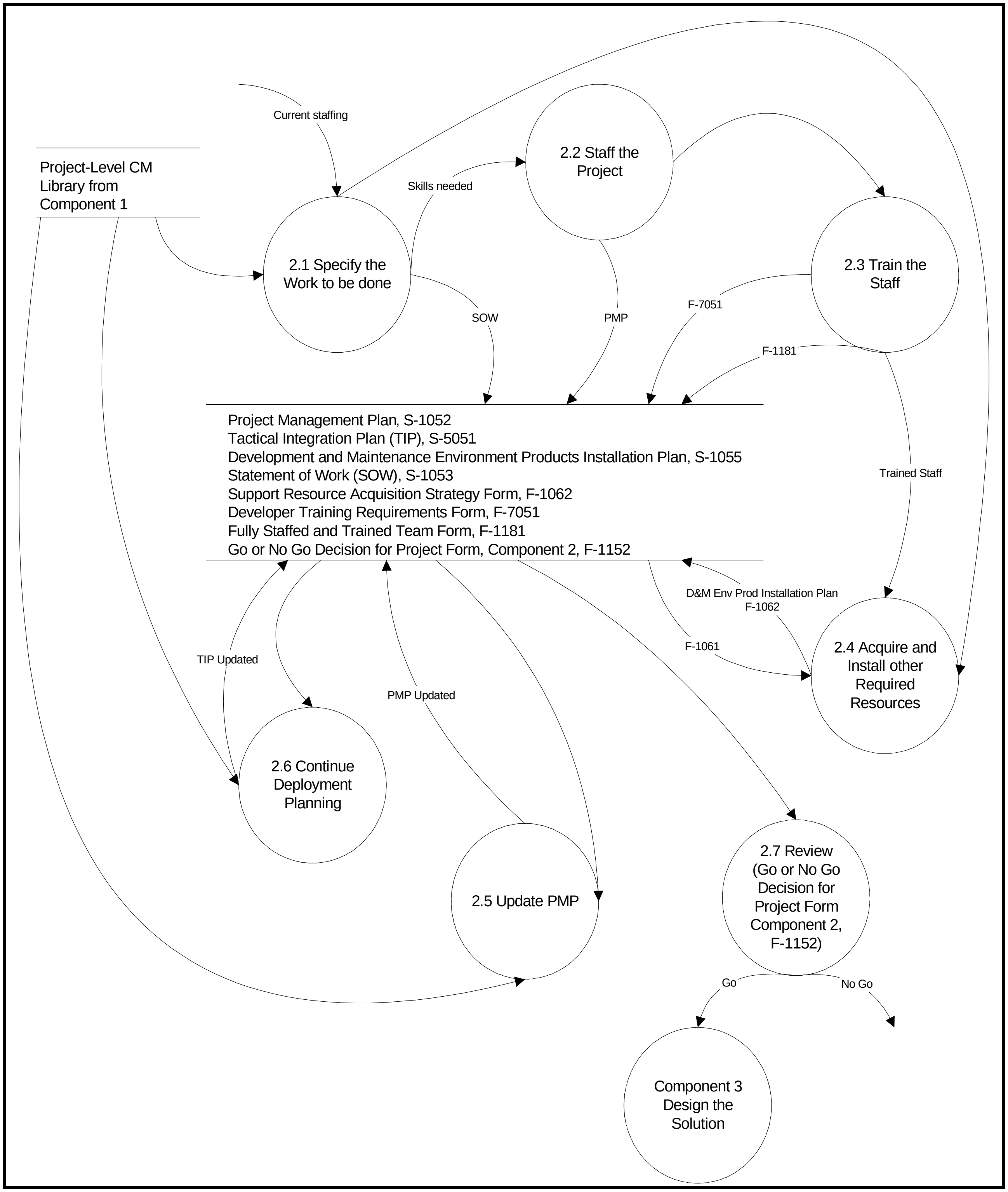

Current staffing
Project-Level CM Library from Component 1
Skills needed
2.2 Staff the Project
2.1 Specify the Work to be done
2.3 Train the Staff
SOW
PMP
F-7051
F-1181
Project Management Plan, S-1052
Tactical Integration Plan (TIP), S-5051
Development and Maintenance Environment Products Installation Plan, S-1055
Statement of Work (SOW), S-1053
Support Resource Acquisition Strategy Form, F-1062
Developer Training Requirements Form, F-7051
Fully Staffed and Trained Team Form, F-1181
Go or No Go Decision for Project Form, Component 2, F-1152
Trained Staff
D&M Env Prod Installation Plan
F-1062
F-1061
2.4 Acquire and Install other Required Resources
TIP Updated
PMP Updated
2.6 Continue Deployment Planning
2.5 Update PMP
2.7 Review (Go or No Go Decision for Project Form Component 2, F-1152)
Go
No Go
Component 3 Design the Solution

Figure 7–2. Component 2

Activity 2.1: Specify the Work to Be Done

See Figure 7–3. Using the products of Component 1 as Input, prepare a statement of work and submit it to a contractor using a contractual funding vehicle. (As explained in Section 7.1, discussion of contracts and funding vehicles is outside the scope of the SDLCM Methodology.)

1. Create a work breakdown structure that specifies the work to be done by both NRC staff and contractors. Develop a matrix showing what work needs to be done and by whom. See Table 7–2.
2. Using the work breakdown structure, create a Statement of Work for contracted resources.

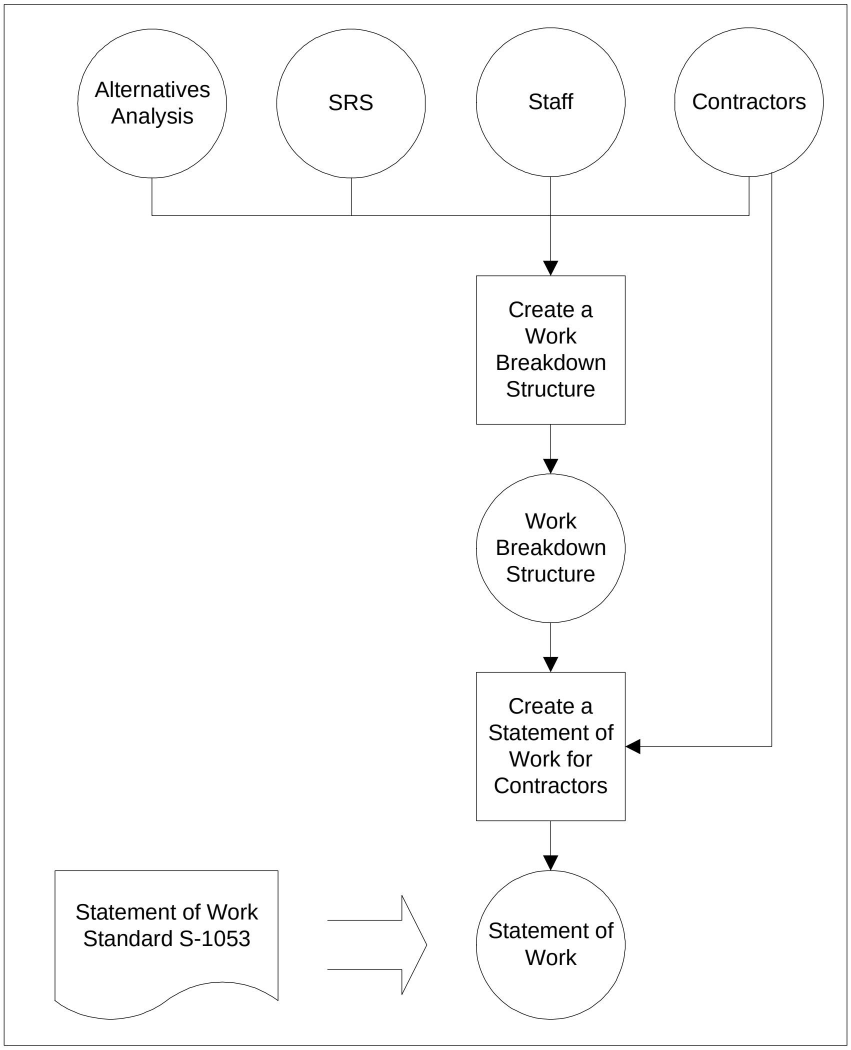

Alternatives Analysis
SRS
Staff
Contractors
Create a Work Breakdown Structure
Work Breakdown Structure
Create a Statement of Work for Contractors
Statement of Work Standard S-1053
Statement of Work

Figure 7–3. Specify the Work to be Done

Table 7–2. Work Matrix

NRC Staff	Contractors	Work to be Done
X		Task 1
	X	Task 2
X		Task *N*

Activity 2.2: Staff the Project

See Figure 7–4. Using the SRS (for an understanding of the technical skills of the people required), an awareness of the current staff, planned staff, and budget:

1. Determine any additional staffing requirements needed for the Project to be a success.
2. Acquire additional staff, using contractors as needed and budgeted for
3. Establish team rules of operation and document them in the Project Management Plan

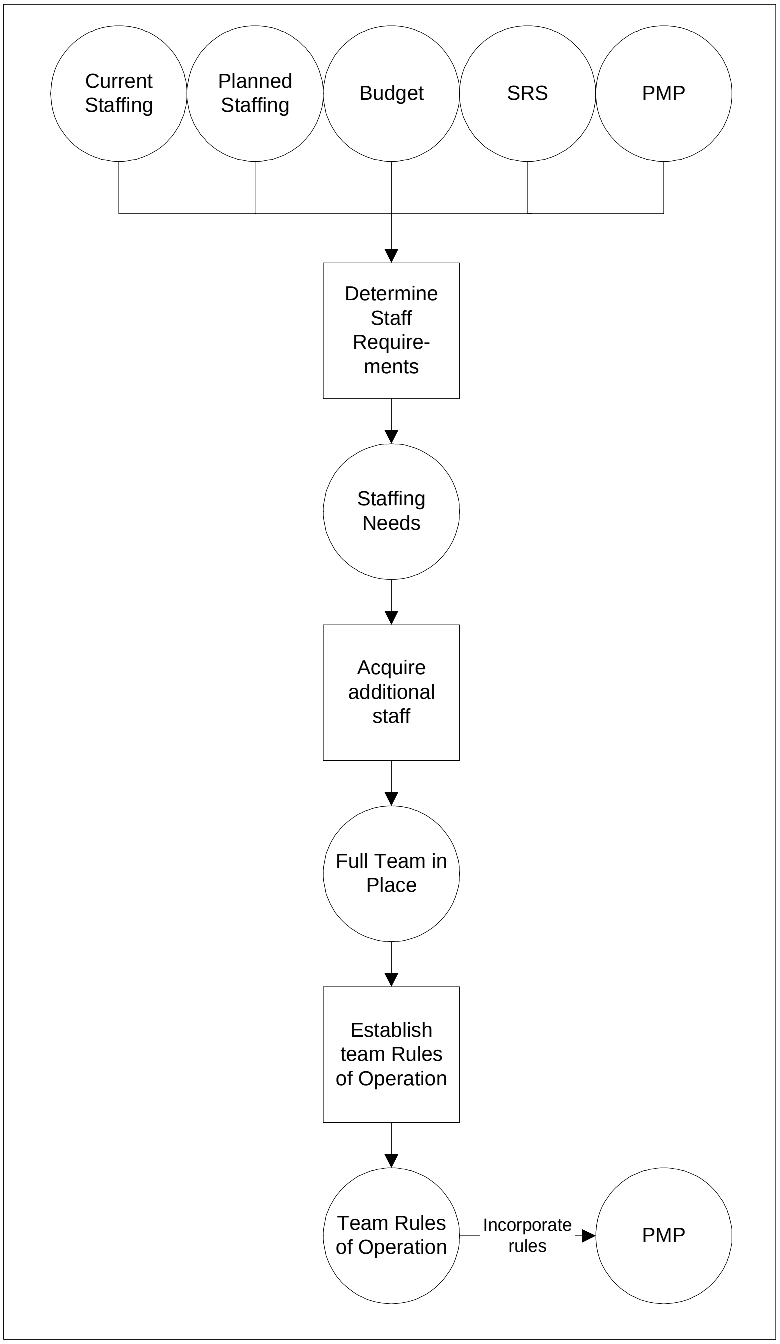


Figure 7–4. Staff the Project

Activity 2.3: Train the Staff

See Figure 7–5. Given a staff with existing skills, and a knowledge of the kinds of skills needed (based on an analysis of the SRS):

1. Determine what training is required
2. Procure the training
3. Perform the training

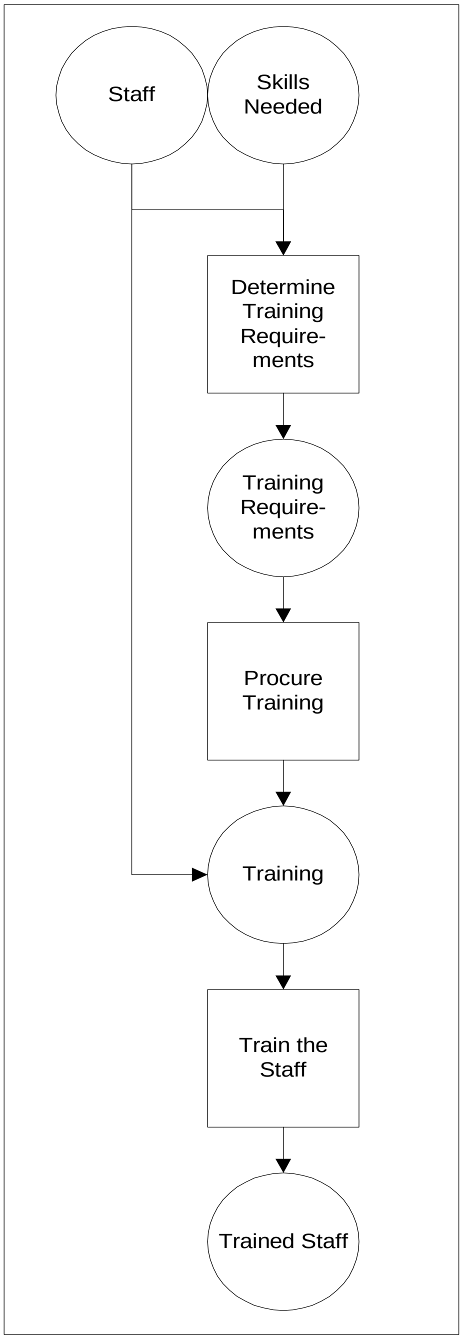


Figure 7–5. Train the Staff

Activity 2.4: Acquire and Install Other Required Resources

See Figure 7–6. Using the Support Resources Acquisition Commitments obtained from Component 1, as well as the Advanced Procurement Plan, budget information, and Tool Request forms:

1. Select an acquisition strategy for additional resources
2. Obtain items within budget
3. Prepare a plan to install resources needed for the development and maintenance environment
4. Install the support resources

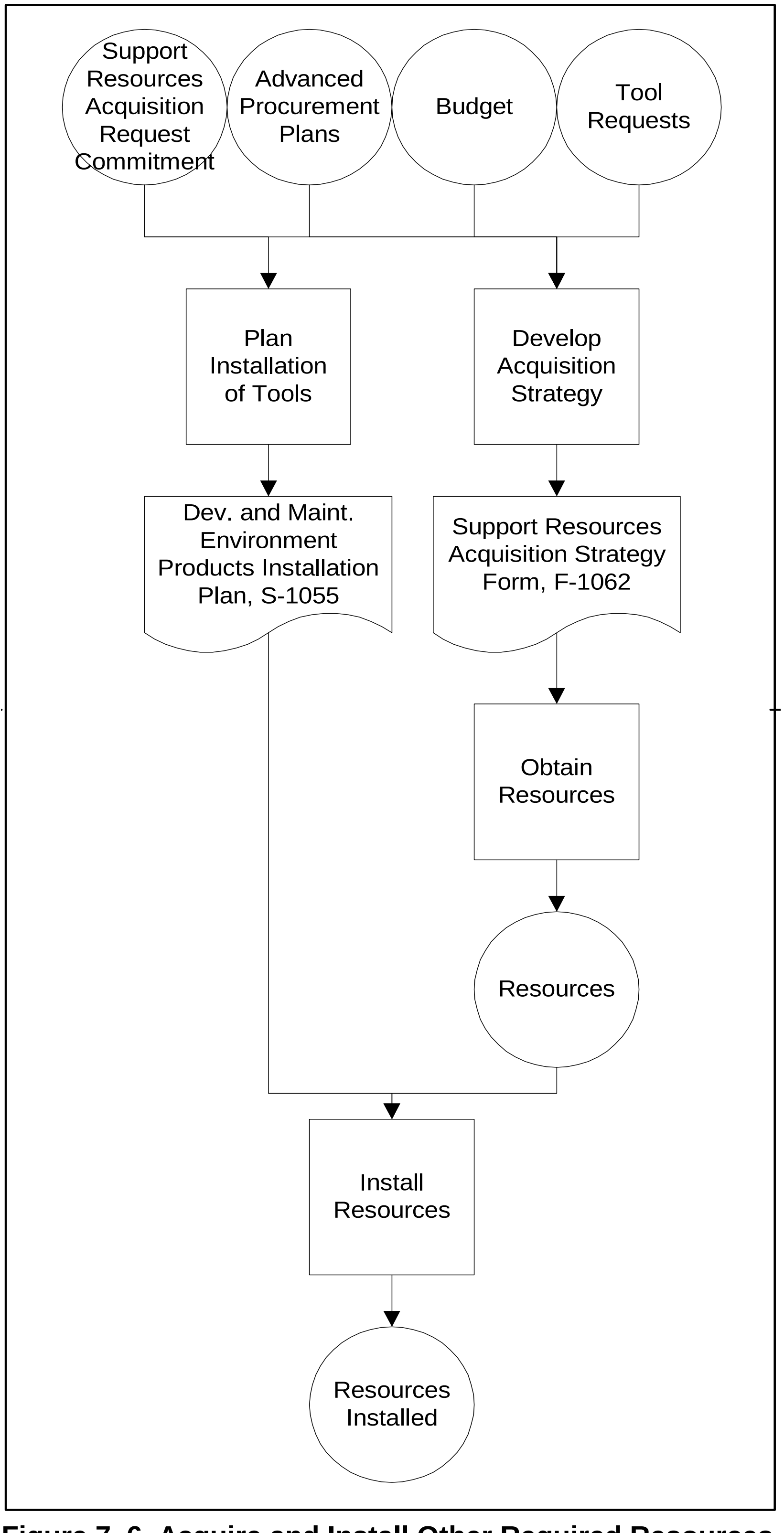

Support Resources Acquisition Request Commitment
Advanced Procurement Plans
Budget
Tool Requests
Plan Installation of Tools
Develop Acquisition Strategy
Dev. and Maint. Environment Products Installation Plan, S-1055
Support Resources Acquisition Strategy Form, F-1062
Obtain Resources
Resources
Install Resources
Resources Installed

Figure 7–6. Acquire and Install Other Required Resources

Activity 2.5: Update the Project Management Plan

See Figure 7–7. Using the current status of the Project staff and other resources,

1. Review the Project Management Plan
2. Update the Project Management Plan so that schedules, staffing profiles, risk assessments, and any other management planning matters reflect the current status

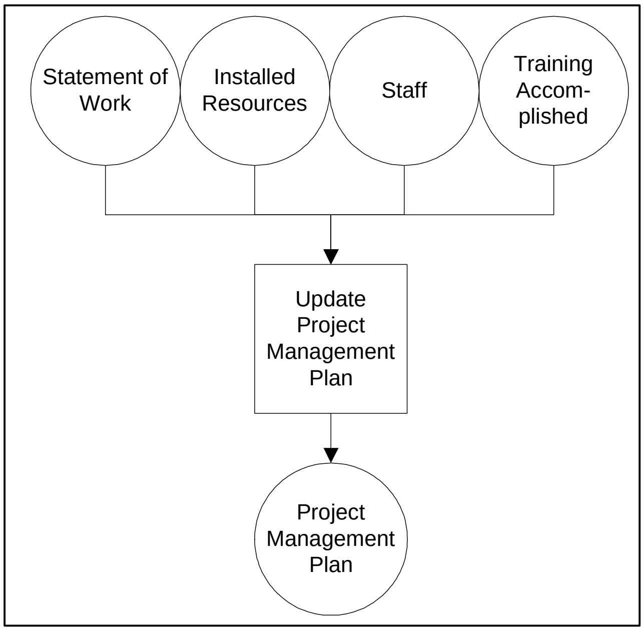


Figure 7–7. Update the Project Management Plan

Activity 2.6: Continue Deployment Planning

See Figure 7–8. Using the current status of the Project,

1. Review the Tactical Integration Plan
2. Update the Tactical Integration Plan so that deployment schedules and any other deployment matters reflect the current status

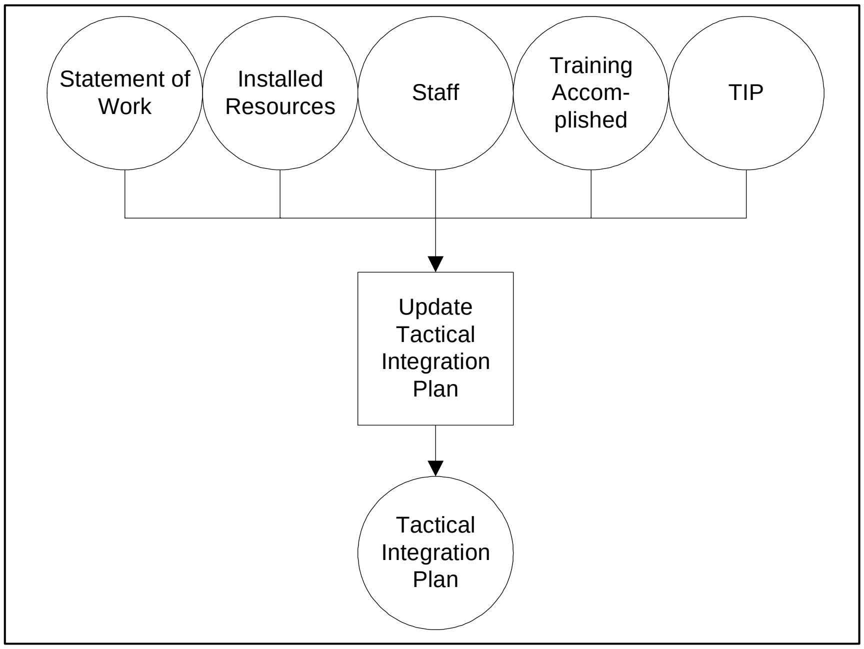


Figure 7–8. Continue Deployment Planning

Activity 2.7: Review (Component 2)

See Figure 7–9. Review the Project to make a GO or NO-GO decision.

1. Review for the need for the Project and the Project status
   - Has the work been specified?
   - Are all equipment resources operational before we move forward with the Project?
   - Is the Project team in place?
   - Is the team trained and ready to begin work?
   - Is there sufficient organizational commitment and executive sponsorship to move forward with the Project?
2. Complete the form required to obtain approval to proceed.

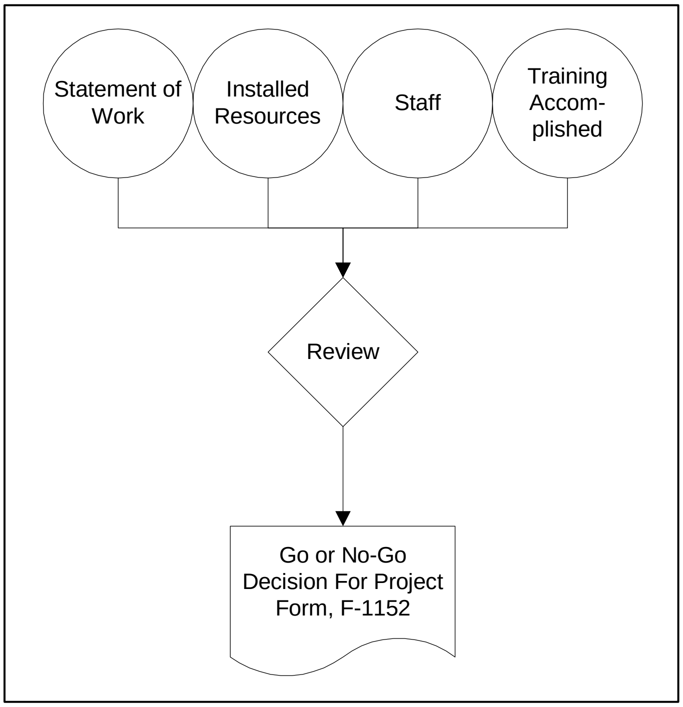


Figure 7–9. Review Component 2

8. Component 3. Design the Solution

8.1 Purpose

The purpose of this component is to:

- Determine functional requirements
- Determine data requirements
- Determine communication requirements
- Determine network requirements
- Document user interface requirements
- Design the solution
- Prepare integration plans
- Prepare Project plans
- Prepare training plans and define operational training materials
- Secure a Go Decision to proceed with engineering the solution

8.2 Roles and Responsibilities

The roles required to perform the activities in the Design the Solution component are shown in Table 8–1.

Table 8–1. Component 3 Roles and Responsibilities

Roles	Responsibilities
Overall Project Manager Business Project Manager Technical Project Manager Development Team	Accountable for Execution
Executive Sponsor Business Advocate	Approval to Proceed
Business Advocate Business SME Technical SME Other Representatives or Stakeholders • Quality Assurance Manager • Configuration Management Manager	Provides Input

8.3 Entry Criteria

The Following events may trigger the initiation of Component 3:

- Completion of Component 2, Acquire Support Resources
- Receipt of additional requirements

8.4 Input, Activities and Outputs

The inputs to this component are as follows:

- Project Management Plan
- Configuration Management Plan
- Quality Assurance Plan
- System Operations Concept
- Current System Assessment Document
- Alternatives Analysis Document
- System Requirements Specification
- Tactical Integration Plan
- Fully Staffed and trained Project team
- Installed resources for design
- Infrastructure
- Selected technology and tools
- Budget allocation
- Enterprise Model
- Technical Reference Model

Figure 8–1 illustrates the flow of the following activities:

1. Analyze Requirements and Perform High Level Design
2. Plan Solution Integration
3. Design Training Materials
4. Establish Test Approach
5. Update Project Management Plan and Software Development Plan
6. Continue Deployment Planning
7. Review (Component 3)

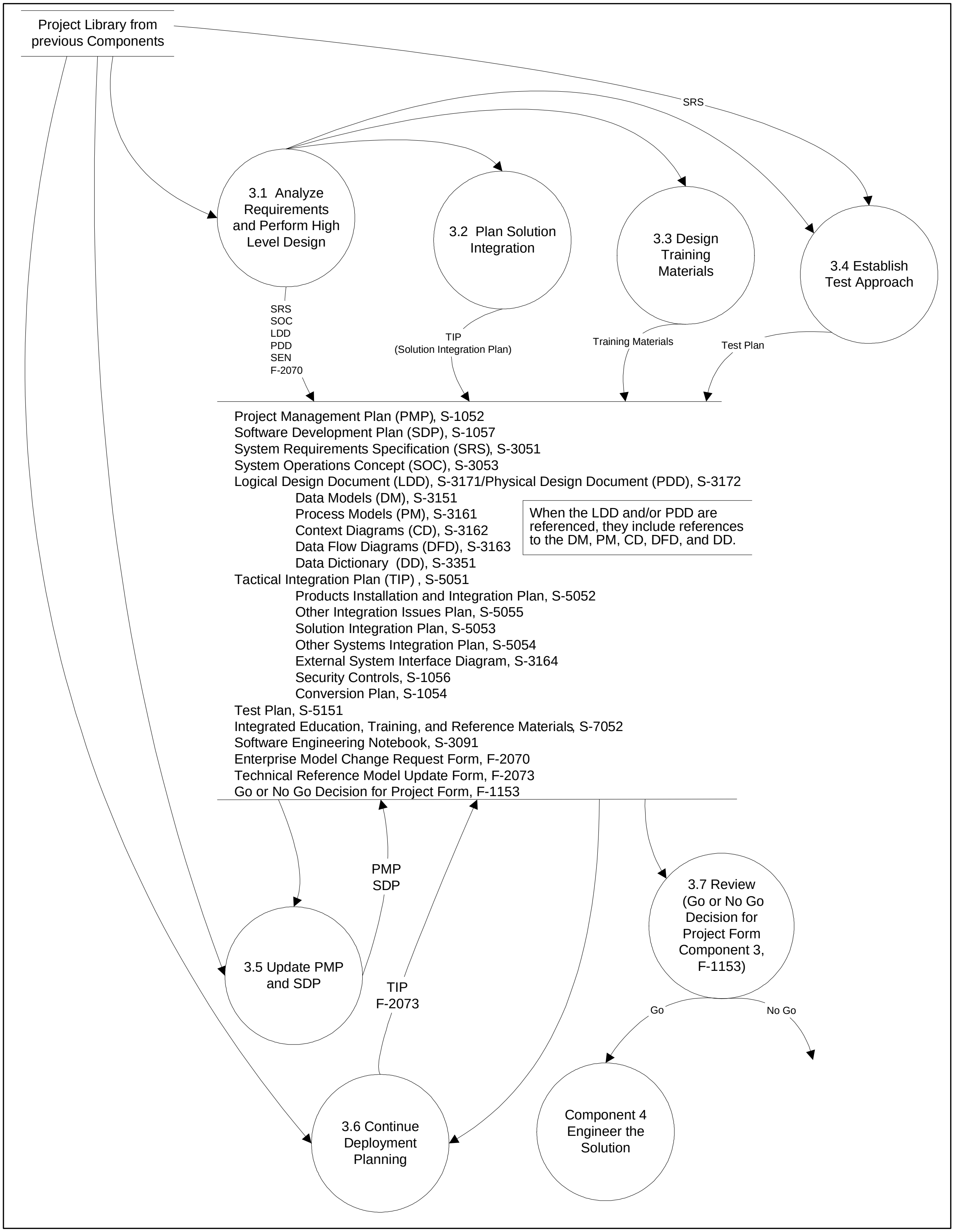

Project Library from previous Components
SRS
3.1 Analyze Requirements and Perform High Level Design
3.2 Plan Solution Integration
3.3 Design Training Materials
3.4 Establish Test Approach
SRS
SOC
LDD
PDD
SEN
F-2070
TIP
(Solution Integration Plan)
Training Materials
Test Plan
Project Management Plan (PMP), S-1052
Software Development Plan (SDP), S-1057
System Requirements Specification (SRS), S-3051
System Operations Concept (SOC), S-3053
Logical Design Document (LDD), S-3171/Physical Design Document (PDD), S-3172
Data Models (DM), S-3151
Process Models (PM), S-3161
Context Diagrams (CD), S-3162
Data Flow Diagrams (DFD), S-3163
Data Dictionary (DD), S-3351
When the LDD and/or PDD are referenced, they include references to the DM, PM, CD, DFD, and DD.
Tactical Integration Plan (TIP) , S-5051
Products Installation and Integration Plan, S-5052
Other Integration Issues Plan, S-5055
Solution Integration Plan, S-5053
Other Systems Integration Plan, S-5054
External System Interface Diagram, S-3164
Security Controls, S-1056
Conversion Plan, S-1054
Test Plan, S-5151
Integrated Education, Training, and Reference Materials, S-7052
Software Engineering Notebook, S-3091
Enterprise Model Change Request Form, F-2070
Technical Reference Model Update Form, F-2073
Go or No Go Decision for Project Form, F-1153
PMP
SDP
3.5 Update PMP and SDP
TIP
F-2073
3.6 Continue Deployment Planning
3.7 Review (Go or No Go Decision for Project Form Component 3, F-1153)
Go
No Go
Component 4 Engineer the Solution

Figure 8–1. Component 3

The inputs to this component were produced as products of earlier components. As Figure 8–1 suggests, many of the activities can be performed in parallel. Figure 8–1 also summarizes the outputs of all Component 3 activities in the central data store. The activities are described in more detail in Section 8.7.

When the Logical Design Document and/or Physical Design Document are referenced in this Handbook, they include references to the Data Models (DM), Process Models (PM), Context Diagrams (CD), Data Flow Diagrams (DFD), and Data Dictionary (DD).

Appendix C includes a summary description of all products of NRC Projects, indicates within which components each product is created and updated, and specifies which products are required.

The products, whose standard and form numbers are shown in the figure, are described in more detail in the companion volume *SDLCM Methodology Procedures, Standards, and Forms*.

8.5 Techniques and Tools

The activities of Component 3 are supported by the following techniques.

- Data Modeling
- Joint Application Design and Development
- Process Modeling
- Prototyping
- Structured Facilitation
- Structured Walkthrough
- Peer Review

Refer to the current *SDLCM Methodology Tool Inventory* for the recommended set of tools.

8.6 Exit Criteria

Component 3 is complete when:

- All critical products have been reviewed (structured walkthrough or peer review).
- Key products have been inspected by QA to conform to Project standards.
- Products have been placed under CM control.
- Information has been shared.
- Issues have been resolved.
- There is management buy-in to move forward.
- There are no reasons to stop this Project now.

8.7 Component 3 Activity Details

The following pages provide detailed activities of Component 3, Design the Solution.

Activity 3.1: Analyze Requirements and Perform High Level Design

This activity is divided into six sub-activities described on the following pages:

3.1.1 Analyze Network

3.1.2 Analyze Data

3.1.3 Analyze and Design Processes

3.1.4 Analyze and Design User Interface

3.1.5 Analyze Platform and Operation System

3.1.6 Design for Functional Requirements

Activity 3.1.1: Analyze Network

See Figure 8–2. Using the Project Charter, System Requirements Specification (SRS) and System Operations Concepts (SOC), determine

- What locations must be supported
- What organizations must be supported
- What are the synchronization requirements

Use this information as input to the subsequent process analysis sub-activity.

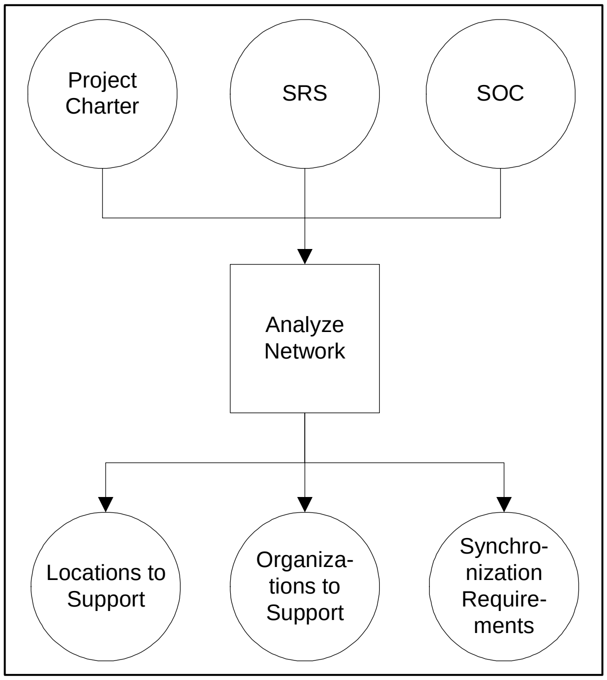


Figure 8–2. Analyze Network

Activity 3.1.2: Analyze Data

See Figure 8–3. Using the Technical Reference Model, Enterprise Model, current version of the System Requirements Specification (which includes the entity list and definitions) and System Operations Concept, perform a more detailed data analysis as follows:

1. Identify all data requirements:
    - Identify corporate data that can be reused verbatim
    - Identify corporate data that can be reused once converted
    - Identify data that must be added
2. From this analysis, create the logical design
    - Specify entity types
    - Specify attribute types
    - Specify entity-relationships
3. Once all entities, attributes, and relationships are established, normalize the database to eliminate redundancy and optimize maintainability. Include Data Models and Data Dictionary.
4. Create the physical design:
    - Map entity types to relational tables
    - Map attribute types to columns
    - Map relationship types to relational schema relationships
5. Calculate the entity volumes.
6. From the entity volumes, compute the data capacity and archiving requirements.
7. Establish security requirements.
8. Select design validation approach.
9. Validate the design.
10. Update the Enterprise Model.

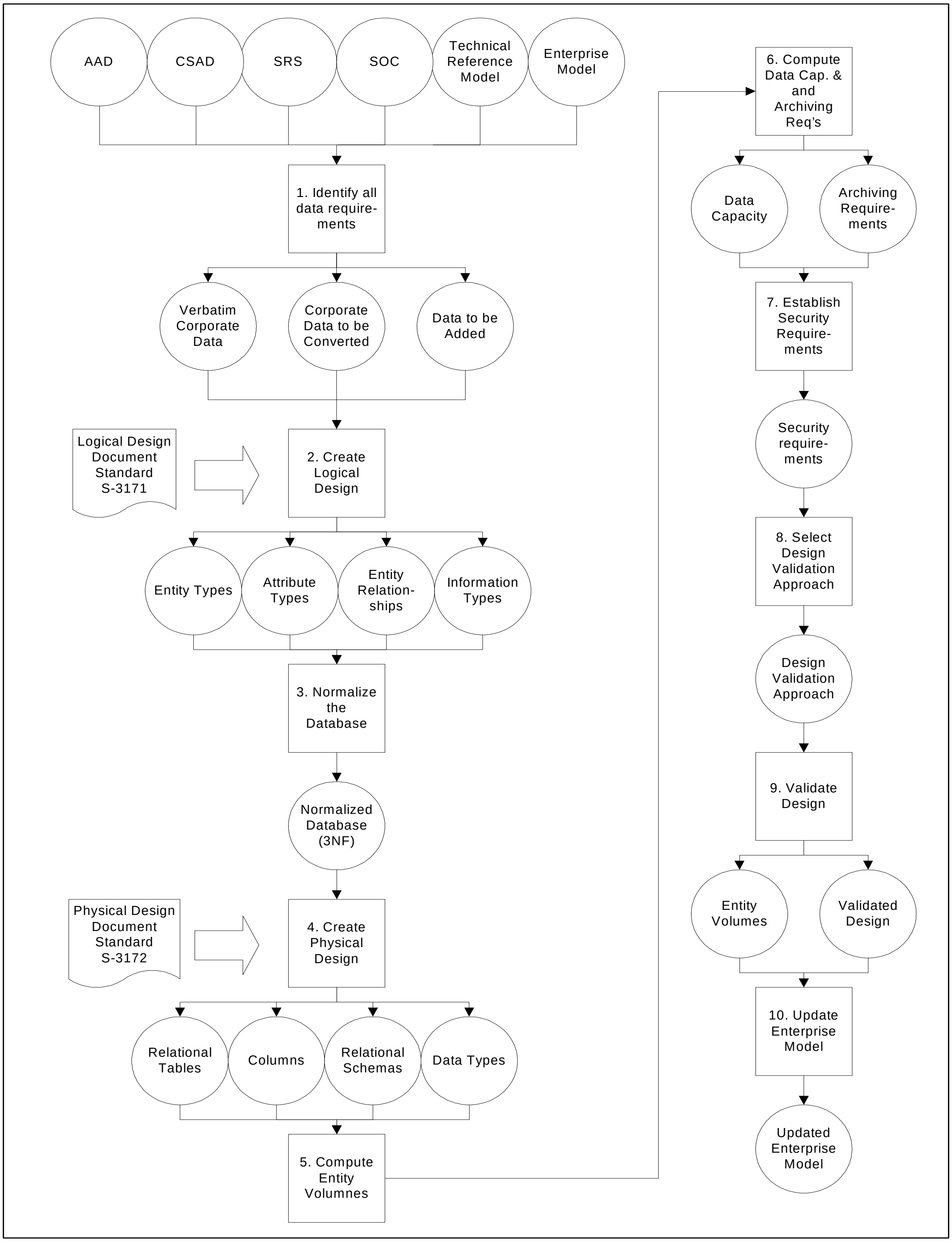

AAD
CSAD
SRS
SOC
Technical Reference Model
Enterprise Model
1. Identify all data requirements
Verbatim Corporate Data
Corporate Data to be Converted
Data to be Added
Logical Design Document Standard S-3171
2. Create Logical Design
Entity Types
Attribute Types
Entity Relationships
Information Types
3. Normalize the Database
Normalized Database (3NF)
Physical Design Document Standard S-3172
4. Create Physical Design
Relational Tables
Columns
Relational Schemas
Data Types
5. Compute Entity Volumnes
6. Compute Data Cap. & and Archiving Req's
Data Capacity
Archiving Requirements
7. Establish Security Requirements
Security requirements
8. Select Design Validation Approach
Design Validation Approach
9. Validate Design
Entity Volumes
Validated Design
10. Update Enterprise Model
Updated Enterprise Model

Figure 8–3. Analyze Data

Activity 3.1.3 Analyze and Design Processes

See Figure 8–4. Using the System Operations Concept and the results of the previous Data Analysis sub-activity, analyze the processes required for the information management system and identify what processes should be automated and which ones should be manual. Document the results of the analysis and design in the logical and physical design documents. Specifically:

1. Identify business procedures, practices, and decision-making activities that must be a part of the information management system. Include the records management requirements of the business.
2. Identify security requirements.
3. Identify external events.
4. Merge the business procedures, security requirements, and external events with the software and hardware part of the information management system to create a data flow diagram. Update Context Diagrams and Data Flow Diagrams, as appropriate.
5. Identify complex processes that must be decomposed and modeled. Create Process Models.
6. Decompose and model complex processes.
7. From the processes, determine which processes should be manual and which should be automated.
8. Establish performance requirements by type of transaction (volume, speed, frequency, etc.)

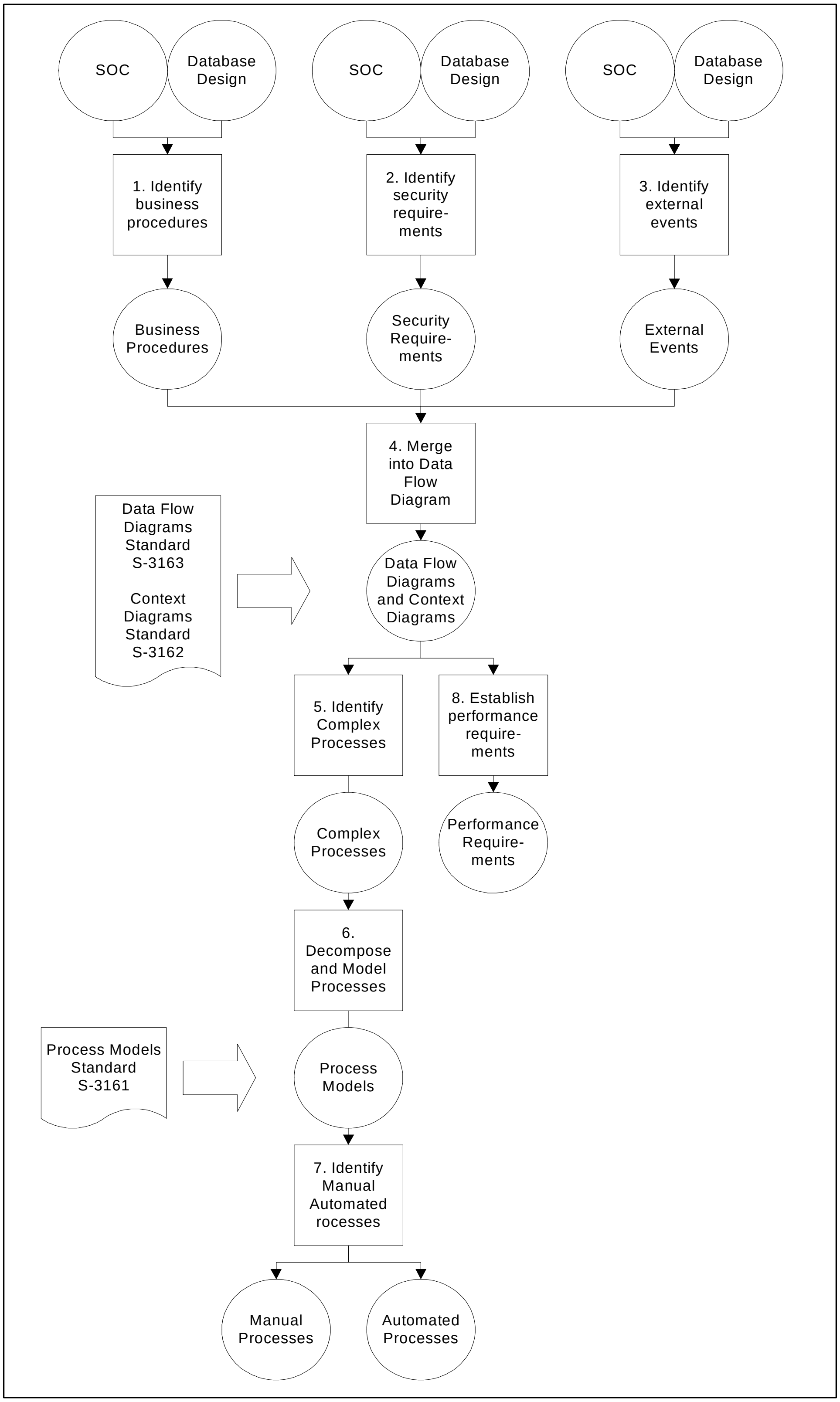

SOC
Database Design
SOC
Database Design
SOC
Database Design
1. Identify business procedures
2. Identify security require-ments
3. Identify external events
Business Procedures
Security Require-ments
External Events
4. Merge into Data Flow Diagram
Data Flow Diagrams Standard S-3163
Context Diagrams Standard S-3162
Data Flow Diagrams and Context Diagrams
5. Identify Complex Processes
8. Establish performance require-ments
Complex Processes
Performance Require-ments
6. Decompose and Model Processes
Process Models Standard S-3161
Process Models
7. Identify Manual Automated rocesses
Manual Processes
Automated Processes

Figure 8–4. Analyze and Design Processes

Activity 3.1.4: Analyze and Design User Interface

See Figure 8–5. Using the results of the database design and process design from previous sub-activities, design the user interface as follows:

1. Determine data entry requirements
2. Determine display requirements
3. Determine dialog requirements
4. Determine screen requirements
5. Create screen prototypes
6. Establish performance requirements

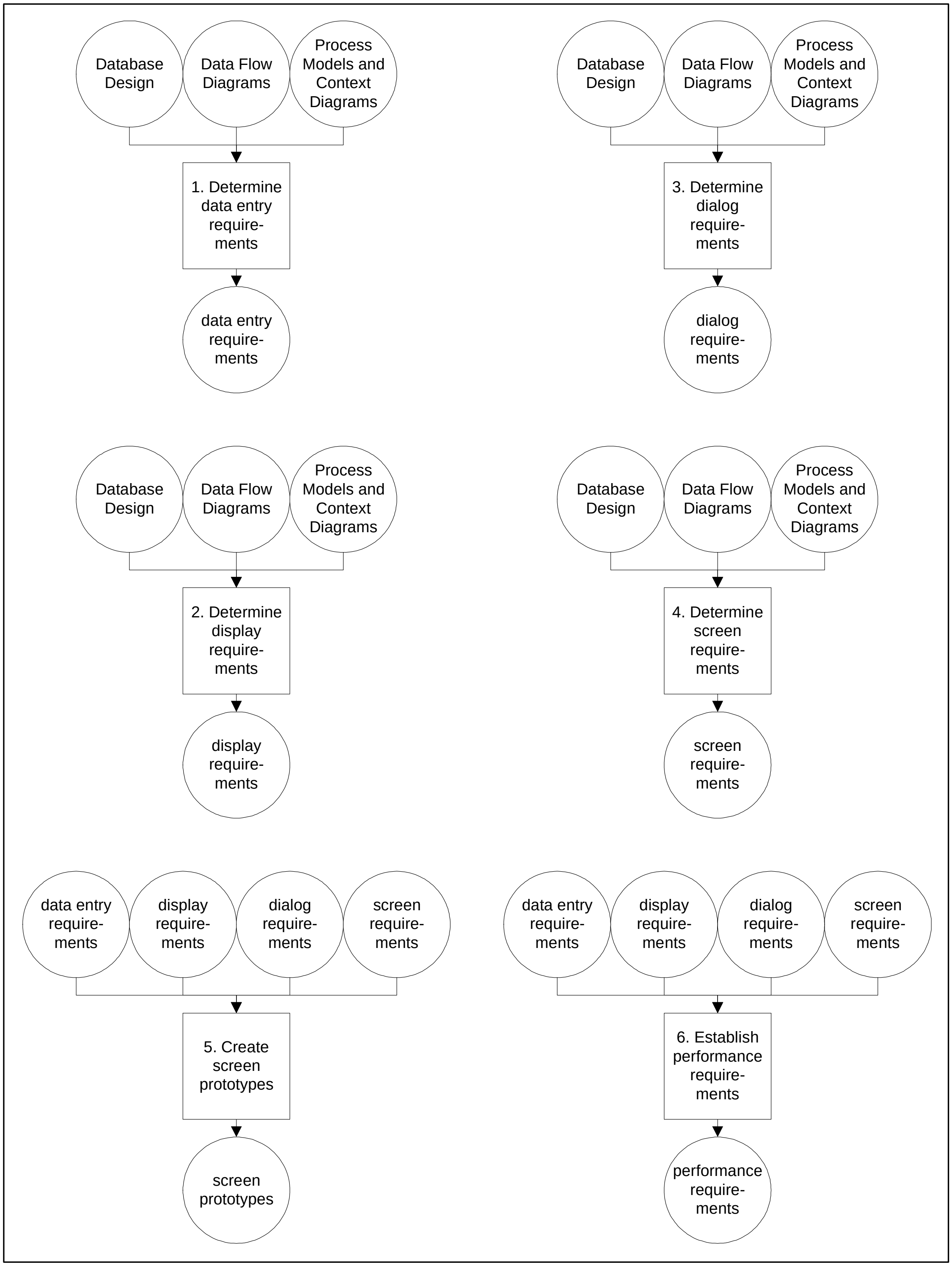


Figure 8–5. Analyze and Design User Interface

Activity 3.1.5: Analyze Platform and Operating System

See Figure 8–6. Using the results of the data analysis, process analysis, and user interface analysis, determine the hardware and operating system requirements.

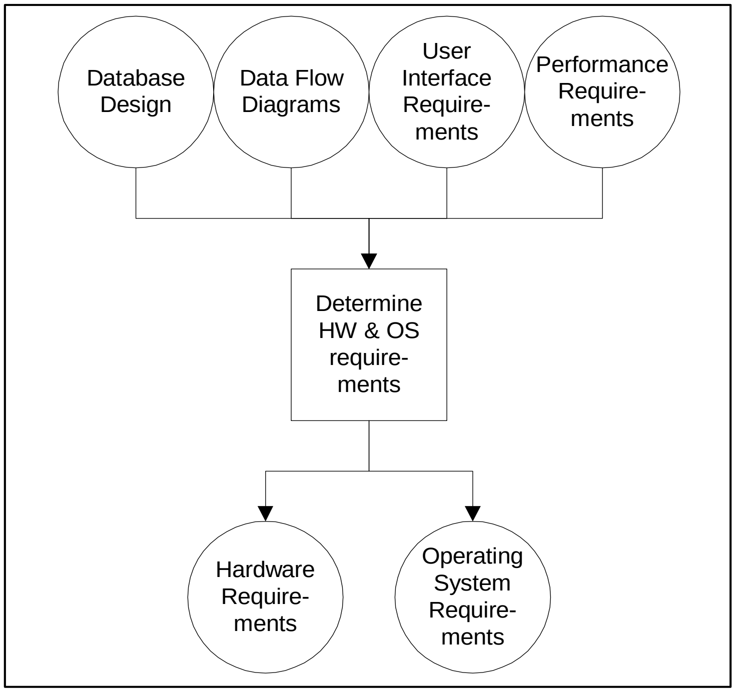

Database Design
Data Flow Diagrams
User Interface Require-ments
Performance Require-ments
Determine HW & OS require-ments
Hardware Require-ments
Operating System Require-ments

Figure 8–6. Analyze Platform and Operation System

Activity 3.1.6: Design for Functional Requirements

See Figure 8–7. Using the results of the:

- Data analysis and database design
- Process analysis and design
- User interface analysis and design
- Hardware and operating system analysis

Design any additional functions required to create an information management system that:

- Reuses corporate databases
- Converts corporate data as required
- Provides support for automating processes appropriately
- Supports the user interfaces as designed above

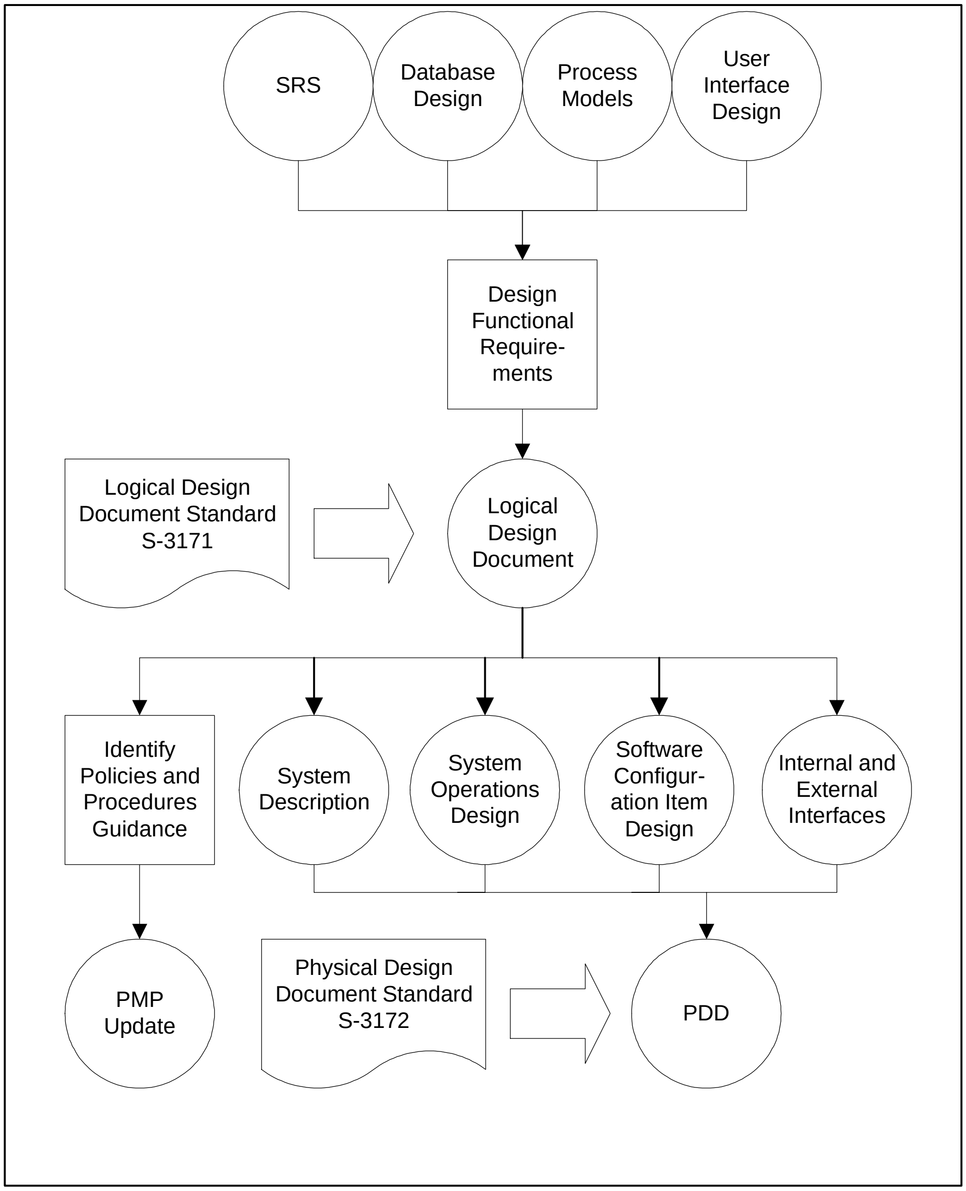

SRS
Database Design
Process Models
User Interface Design
Design Functional Require-ments
Logical Design Document Standard S-3171
Logical Design Document
Identify Policies and Procedures Guidance
System Description
System Operations Design
Software Configur-ation Item Design
Internal and External Interfaces
PMP Update
Physical Design Document Standard S-3172
PDD

Figure 8-7. Design for Functional Requirements

Activity 3.2: Plan Solution Integration

See Figure 8–8. Using the results of Activity 3.1, develop a plan for integrating the solution.

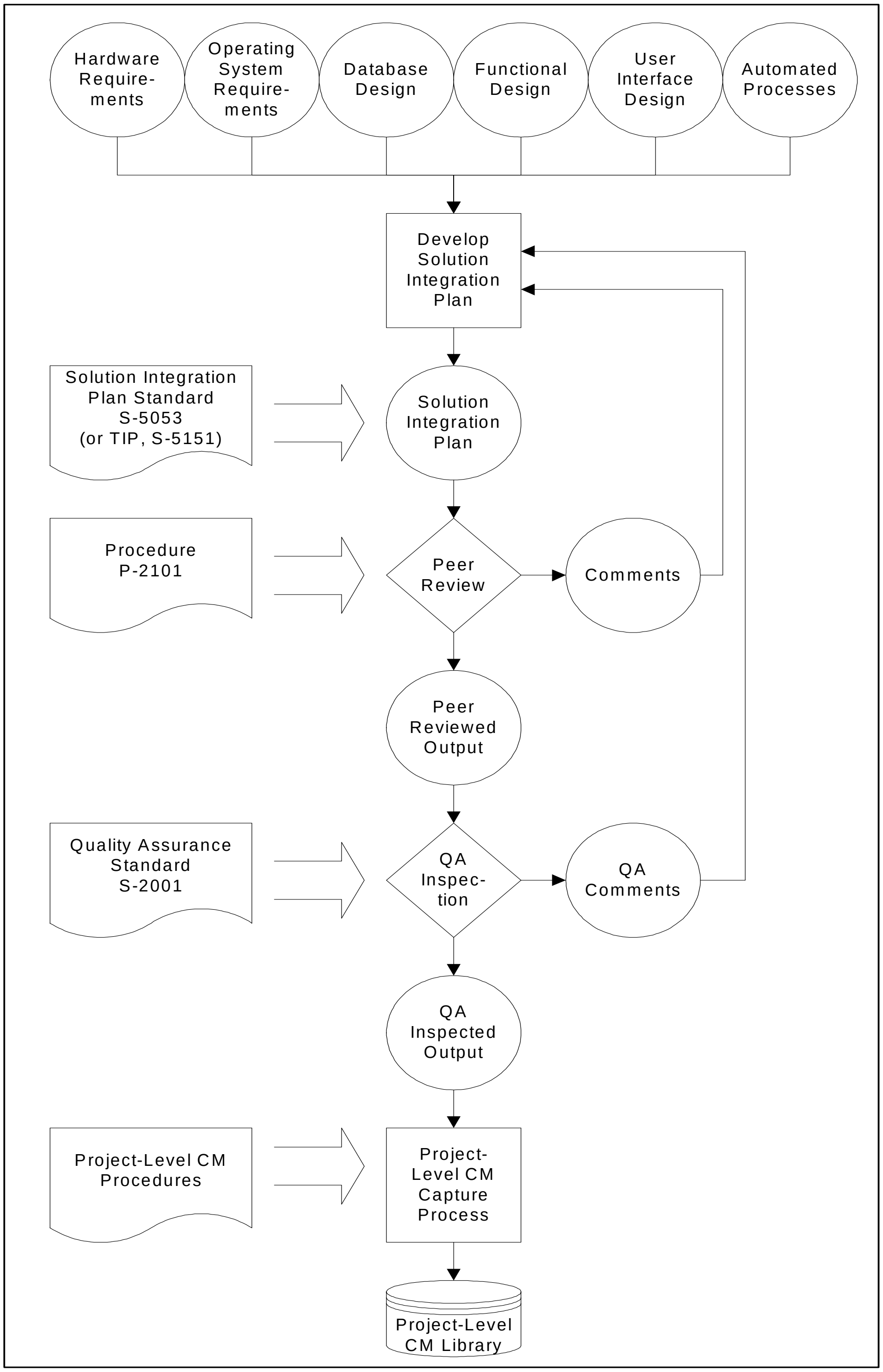

Hardware Requirements
Operating System Requirements
Database Design
Functional Design
User Interface Design
Automated Processes
Develop Solution Integration Plan
Solution Integration Plan Standard S-5053 (or TIP, S-5151)
Solution Integration Plan
Procedure P-2101
Peer Review
Comments
Peer Reviewed Output
Quality Assurance Standard S-2001
QA Inspection
QA Comments
QA Inspected Output
Project-Level CM Procedures
Project-Level CM Capture Process
Project-Level CM Library

Figure 8–8. Plan Solution Integration

Activity 3.3: Design Training Materials

See Figure 8–9. Using the system requirements and the results of requirements analysis and high level design, design the training materials.

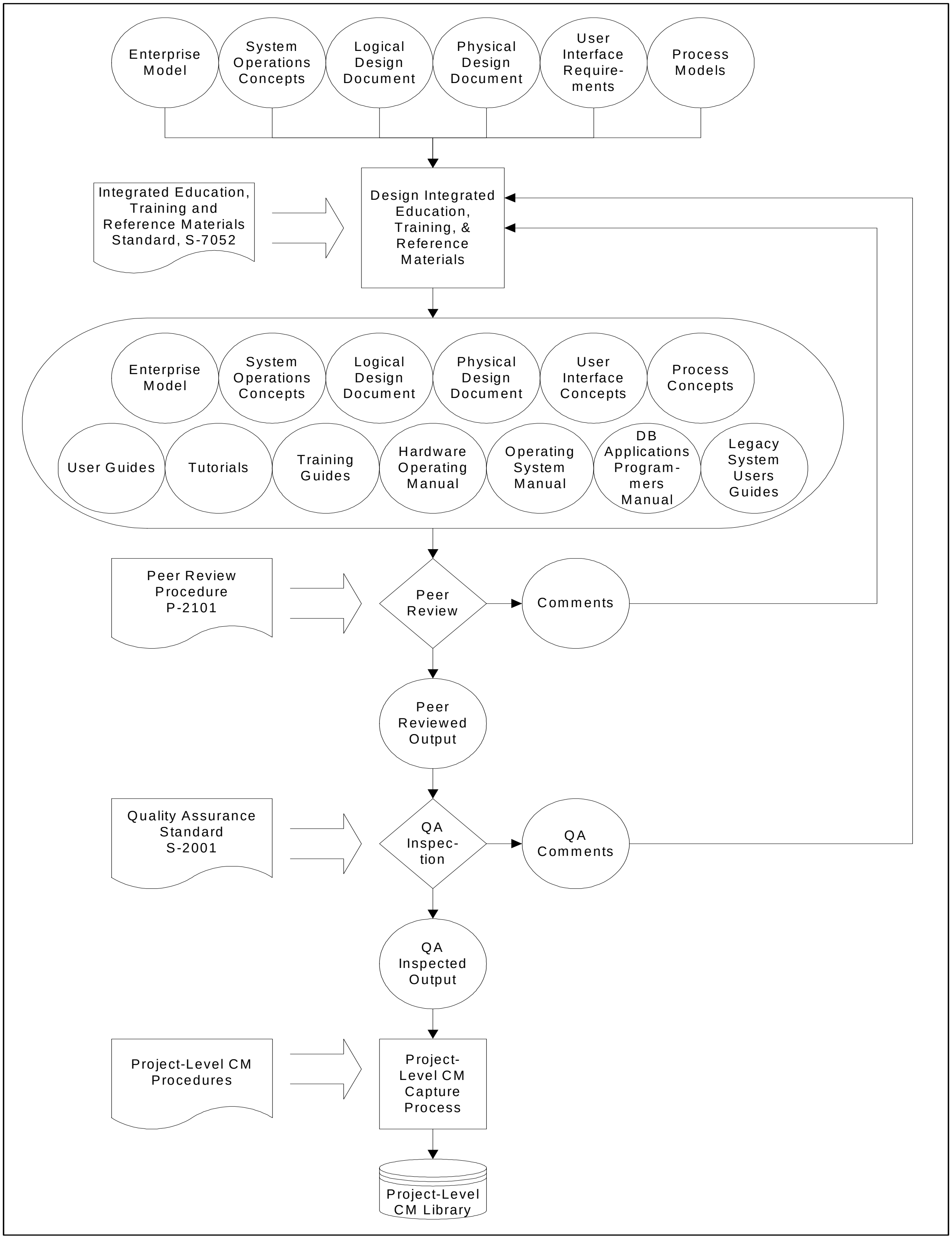

Enterprise Model
System Operations Concepts
Logical Design Document
Physical Design Document
User Interface Require-ments
Process Models
Integrated Education, Training and Reference Materials Standard, S-7052
Design Integrated Education, Training, & Reference Materials
Enterprise Model
System Operations Concepts
Logical Design Document
Physical Design Document
User Interface Concepts
Process Concepts
User Guides
Tutorials
Training Guides
Hardware Operating Manual
Operating System Manual
DB Applications Program-mers Manual
Legacy System Users Guides
Peer Review Procedure P-2101
Peer Review
Comments
Peer Reviewed Output
Quality Assurance Standard S-2001
QA Inspec-tion
QA Comments
QA Inspected Output
Project-Level CM Procedures
Project-Level CM Capture Process
Project-Level CM Library

Figure 8–9. Design Training Materials

Activity 3.4: Establish Test Approach

See Figure 8–10. Using the results of the requirements analysis and high level design, including knowledge of the:

- Top-level requirements (functions)
- Systems operations concepts
- Systems integration plan
- New system functions and databases
- Computer software configuration items (CSCIs)
- Units

Specify a bottom-up test approach, which must contain:

- Unit Tests
- Module Tests
- New Integrated Solution Test
- Systems Test
- Requirements Test

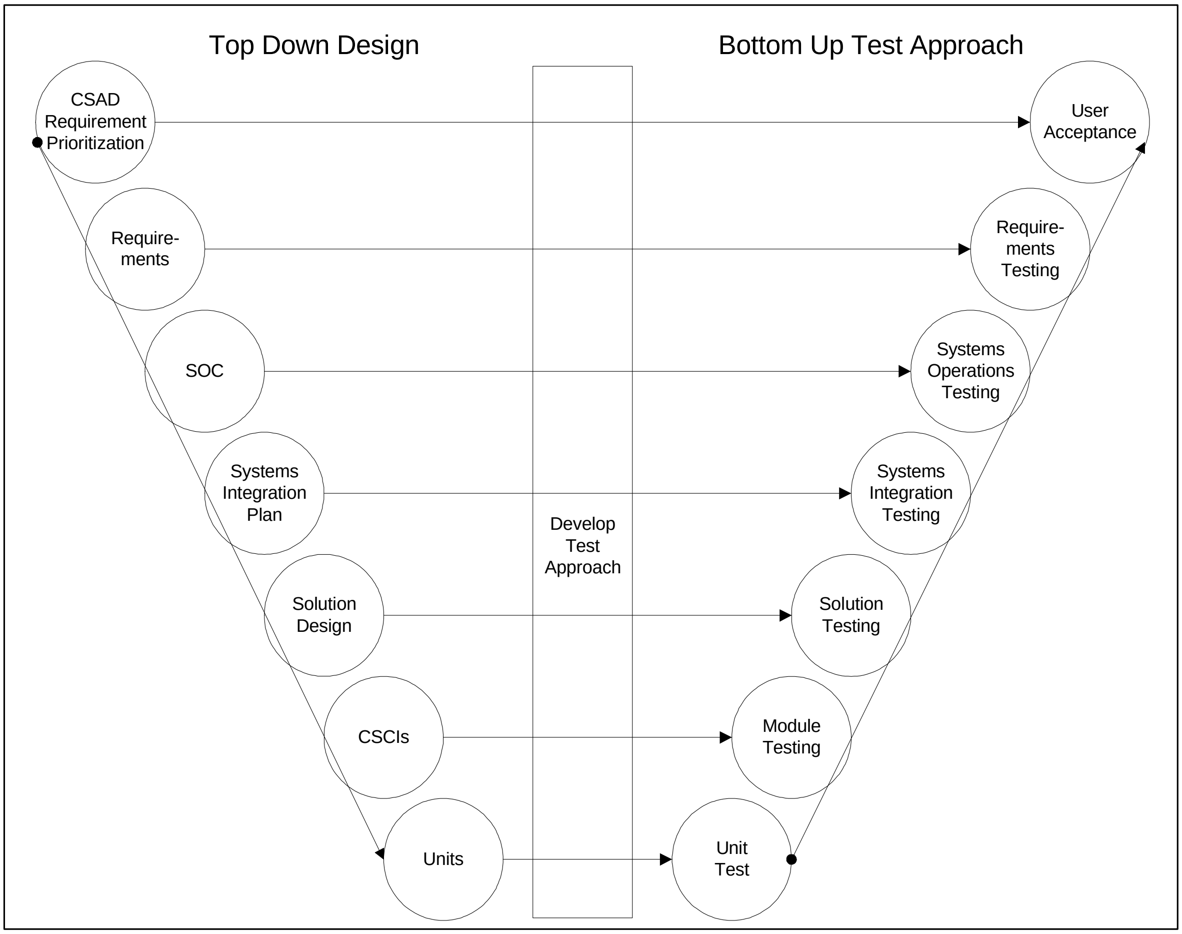


Figure 8–10. Establish Test Approach

Activity 3.5: Update Project Management Plan

See Figure 8–11. Using the work results, identify any changes required in the resources, schedules, and activities. Update the Project Management Plan.

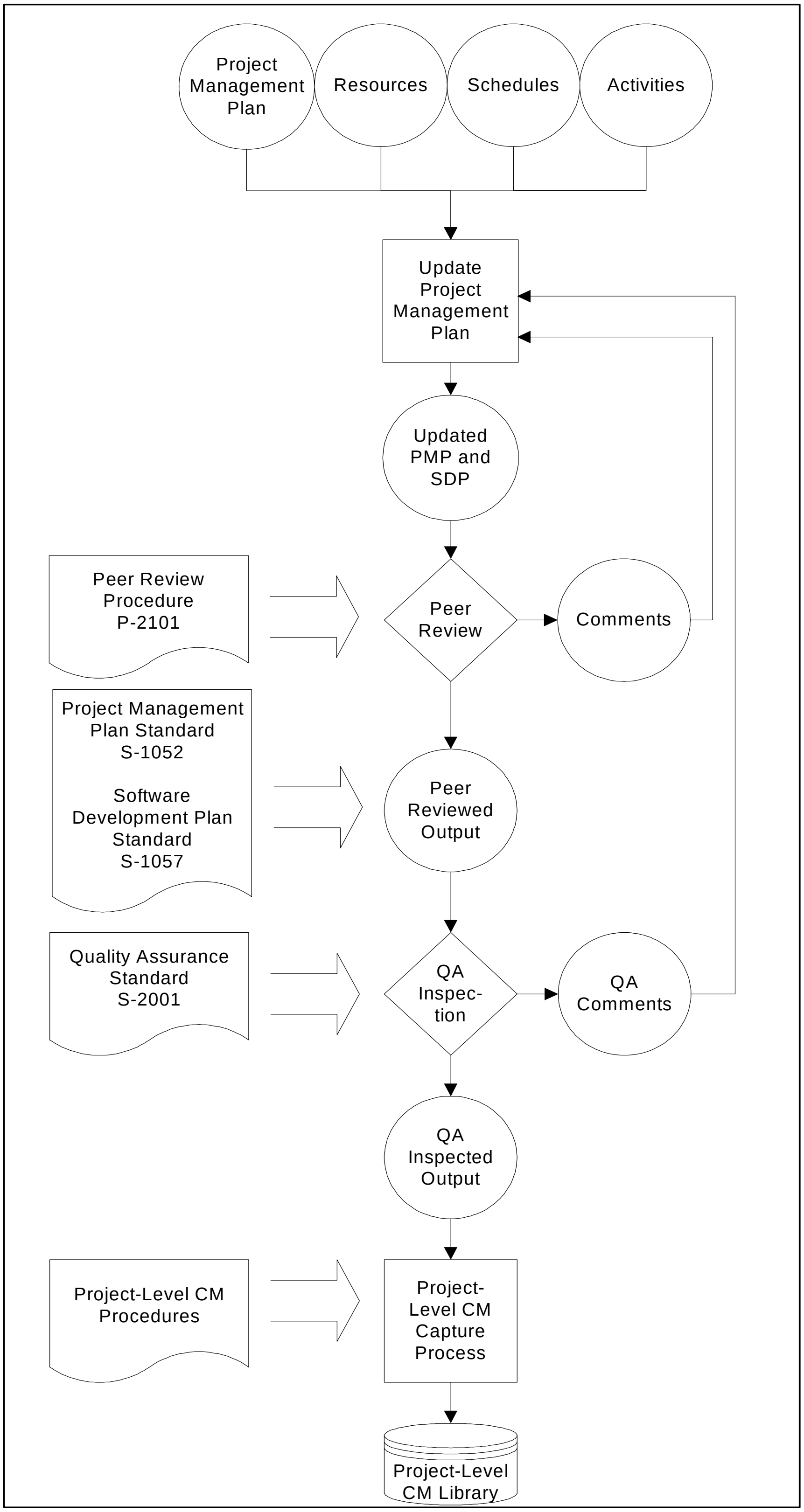

Project Management Plan
Resources
Schedules
Activities
Update Project Management Plan
Updated PMP and SDP
Peer Review Procedure P-2101
Peer Review
Comments
Project Management Plan Standard S-1052
Software Development Plan Standard S-1057
Peer Reviewed Output
Quality Assurance Standard S-2001
QA Inspec-tion
QA Comments
QA Inspected Output
Project-Level CM Procedures
Project-Level CM Capture Process
Project-Level CM Library

Figure 8–11. Update Project Management Plan

Activity 3.6: Continue Deployment Planning

See Figure 8–12. Using the Solution Integration Plan, and the results of the network analysis, analysis of other systems that can be reused verbatim, analysis of other systems that can be reused in part by establishing conversion requirements, establish a system integration plan.

1. Identify other systems for integration
2. Identify systems security requirements
3. Update the other portions of the TIP or development separate sub-plans
4. Update Technical Reference Model

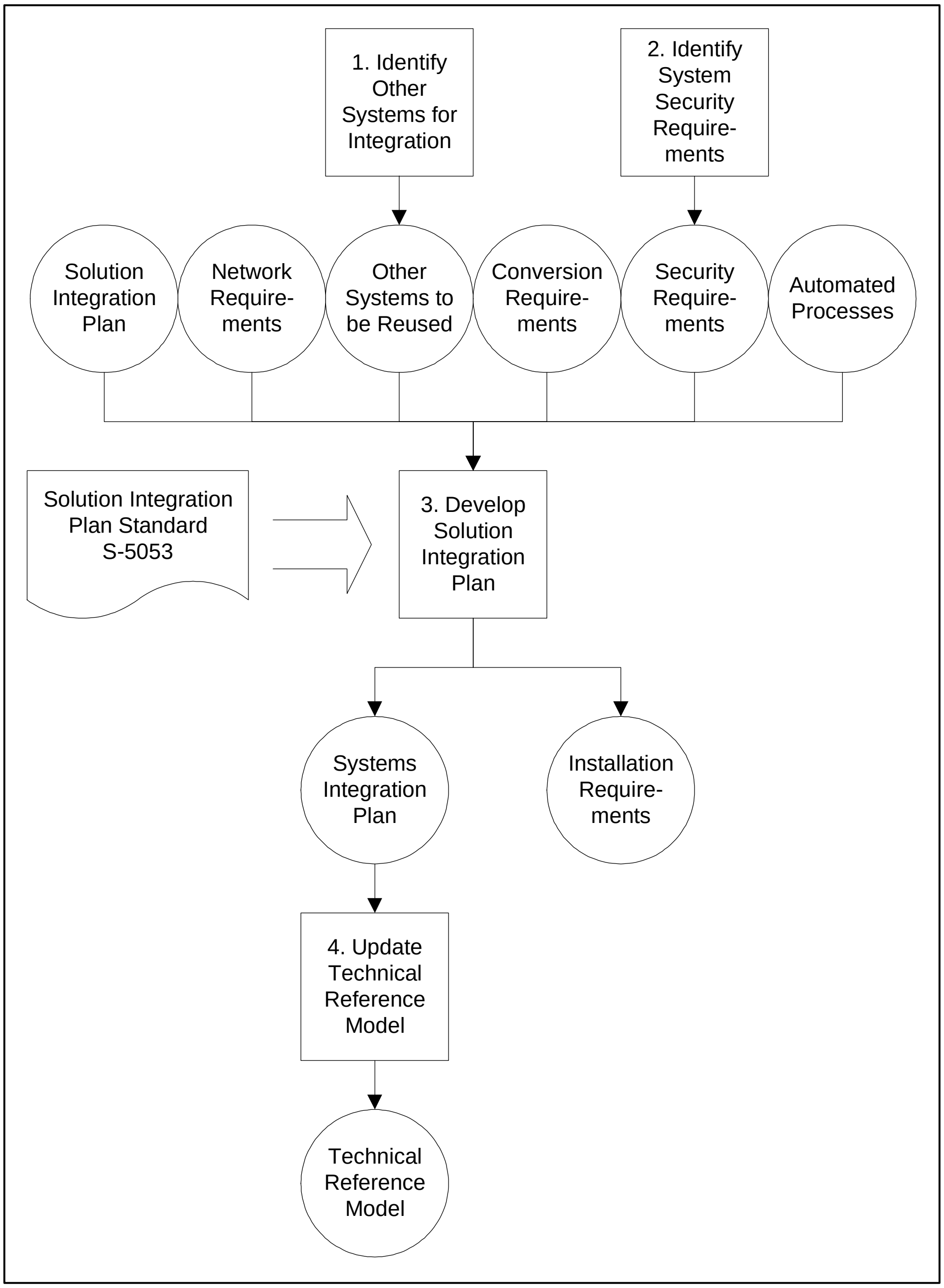

1. Identify Other Systems for Integration
2. Identify System Security Require-ments
Solution Integration Plan
Network Require-ments
Other Systems to be Reused
Conversion Require-ments
Security Require-ments
Automated Processes
Solution Integration Plan Standard S-5053
3. Develop Solution Integration Plan
Systems Integration Plan
Installation Require-ments
4. Update Technical Reference Model
Technical Reference Model

Figure 8–12. Continue Deployment Planning

Activity 3.7: Review (Component 3)

See Figure 8–13. Review all products of the Project to make a GO or NO-GO decision before proceeding to engineer all or a portion of the solution. Ensure that all stakeholders, including the Records Management Branch, are included in the review activities. Note that the LDD and PDD represent all of the Data Flow Diagrams, Context Diagrams, Data Dictionary, Data Models and Process Models.

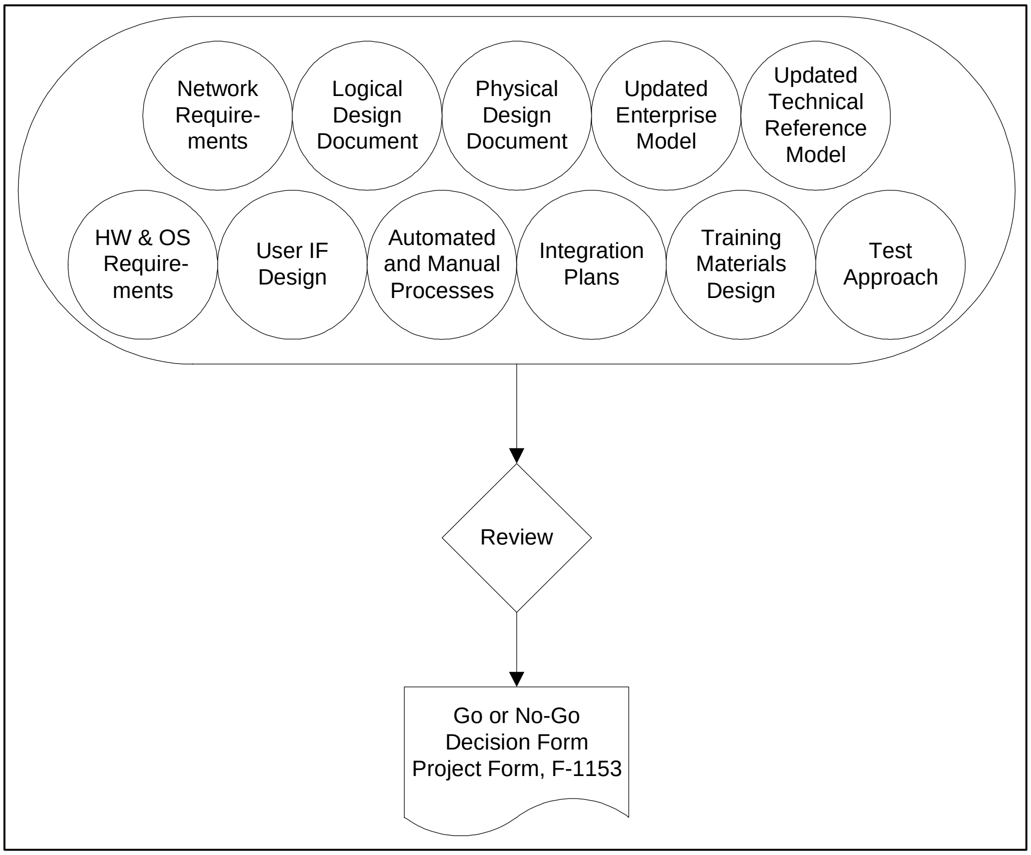


Figure 8–13. Review (Component 3)

9. Component 4. Engineer the Solution

9.1 Purpose

The purpose of this component is to build the application system, which includes the following:

- Maintain the Change Log
- Validate/Refine Design
- Select an engineering approach
- Construct or acquire all modules of the system
- Unit Test
- Integrate the solution
- System test the solution
- Create the rollout strategy and continue planning for deployment
- Assess and update agency policies and procedures
- Construct the policy and procedures that support the system and the required training and education materials
- Update the Project Management Plan
- Secure a Go decision to move to pre-deployment testing

9.2 Roles and Responsibilities

The roles required to perform the activities in the Engineer the Solution component are shown in Table 9–1.

Table 9–1. Component 4 Roles and Responsibilities

Roles	Responsibilities
Overall Project Manager Technical Project Manager Technical SME for Infrastructure Development Team	Accountable for Execution of Activities
Executive Sponsor Business Advocate Business Project Manager Business SME Other Technical SMEs	Provides Input

9.3 Entry Criteria

The following event may trigger the initiation of Component 4:

- Some portion of the solution design approved for engineering

9.4 Input, Activities, and Outputs

The inputs to this component are as follows:

- Updated Project Management Plan
- Updated Systems Requirements Specification
- Logical Design Document
- Physical Design Document
- Updated Tactical Integration Plan, including
    - Products Installation and Integration Plan
    - Other Integration Issues Plan
    - Solution Integration Plan
    - Other Systems Integration Plan
    - Security Plan
    - Conversion Plan
- Test Plan
- Integrated Education, Training, and Reference Materials
- Software Engineering Notebook
- Change Requests, if any

The inputs to this component were produced as products of earlier components.

Figure 9–1 illustrates the flow of the following activities:

1. Maintain the Change Log
2. Engineer the Detailed Design
3. Build the Solution
4. Integrate the Solution
5. Test the Solution
6. Create the Rollout Strategy and Continue Deployment Planning
7. Update Policies and Procedures
8. Build the User Material and Training Materials
9. Deliver Training
10. Update the Project Management Plan and Software Development Plan
11. Review (Component 4) and Transfer Materials to Central Configuration Management

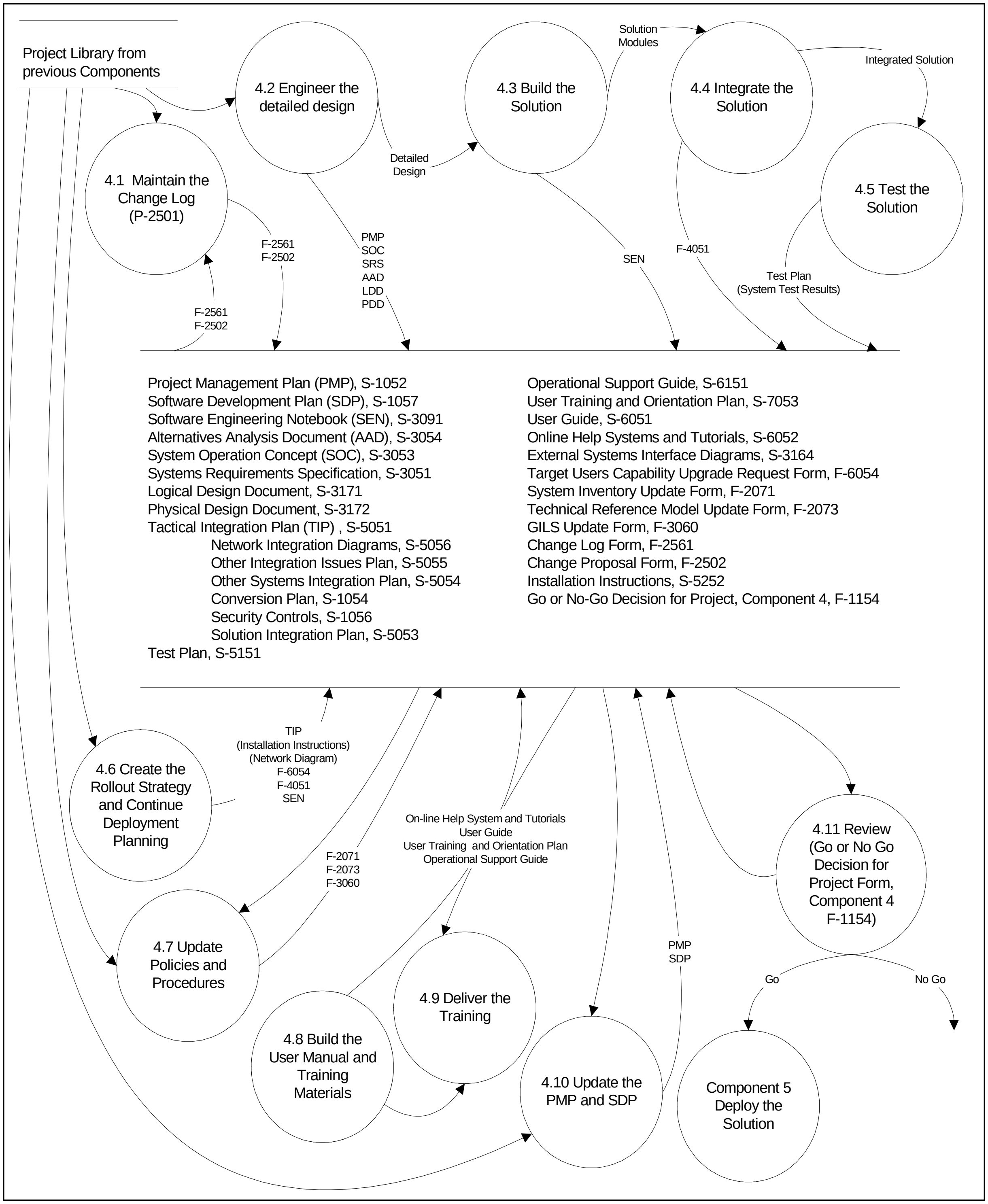

Project Library from previous Components
4.1 Maintain the Change Log (P-2501)
4.2 Engineer the detailed design
4.3 Build the Solution
4.4 Integrate the Solution
4.5 Test the Solution
Solution Modules
Integrated Solution
Detailed Design
F-2561
F-2502
F-2561
F-2502
PMP
SOC
SRS
AAD
LDD
PDD
SEN
F-4051
Test Plan
(System Test Results)
Project Management Plan (PMP), S-1052
Software Development Plan (SDP), S-1057
Software Engineering Notebook (SEN), S-3091
Alternatives Analysis Document (AAD), S-3054
System Operation Concept (SOC), S-3053
Systems Requirements Specification, S-3051
Logical Design Document, S-3171
Physical Design Document, S-3172
Tactical Integration Plan (TIP) , S-5051
Network Integration Diagrams, S-5056
Other Integration Issues Plan, S-5055
Other Systems Integration Plan, S-5054
Conversion Plan, S-1054
Security Controls, S-1056
Solution Integration Plan, S-5053
Test Plan, S-5151
Operational Support Guide, S-6151
User Training and Orientation Plan, S-7053
User Guide, S-6051
Online Help Systems and Tutorials, S-6052
External Systems Interface Diagrams, S-3164
Target Users Capability Upgrade Request Form, F-6054
System Inventory Update Form, F-2071
Technical Reference Model Update Form, F-2073
GILS Update Form, F-3060
Change Log Form, F-2561
Change Proposal Form, F-2502
Installation Instructions, S-5252
Go or No-Go Decision for Project, Component 4, F-1154
TIP
(Installation Instructions)
(Network Diagram)
F-6054
F-4051
SEN
4.6 Create the Rollout Strategy and Continue Deployment Planning
On-line Help System and Tutorials
User Guide
User Training and Orientation Plan
Operational Support Guide
F-2071
F-2073
F-3060
4.7 Update Policies and Procedures
4.8 Build the User Manual and Training Materials
4.9 Deliver the Training
4.10 Update the PMP and SDP
PMP
SDP
4.11 Review (Go or No Go Decision for Project Form, Component 4 F-1154)
Go
No Go
Component 5 Deploy the Solution

Figure 9–1. Component 4

As the figure suggests, some of the activities can be performed in parallel. However, where decisions or data must be presented from one activity to another, the activities must be performed sequentially (i.e., 4.1 through 4.5, 4.6 and 4.8, 4.7 is dependent upon 4.5 and 4.6 for input.). The activities are described in more detail in Section 9.7.

Figure 9–1 also summarizes the outputs of all Component 4 activities in the central data store.

Appendix C includes a summary description of all products of NRC projects, indicates within which components each product is created and updated, and specifies which products are required.

The products, whose standard and form numbers are shown in the figure, are described in more detail in the companion volume *SDLCM Methodology Procedures, Standards, and Forms*.

9.5 Techniques and Tools

The activities of Component 4 are supported by the following techniques:

- Joint Application Design and Development
- Rapid Application Design
- Structured Facilitation
- Structured Walkthrough
- Peer Review

Refer to the current Technical Reference Model, which should have been decided in Component 2.

9.6 Exit Criteria

Component 4 Is complete when:

- All critical software products have been reviewed (structured walkthrough or peer review)
- All critical procedures, guidelines, etc. reflecting changes to business approach and rules have been circulated for comment and concurrence has been received.
- Key products have been inspected by QA to conform to project standards
- Products have been placed under CM control
- Information has been shared
- Issues have been resolved
- There is management buy-in to move forward
- There are no reasons to stop this project now

9.7 Component 4 Activity Details

The following pages provide detailed activities of Component 4, Engineer the Solution.

Activity 4.1: Maintain the Change Log

See Figure 9–2. When any portion of the design is approved for engineering, it is important to maintain a record of any proposed changes (whether required or requested).

1. Initiate the change log. Use SDLCM Methodology Form F–2561 or an automated software tool that provides at least the same information.
2. Process change proposals and maintain the change log following the Change Proposal procedure (P–2502) and using the Change Proposal Form (F–2502).

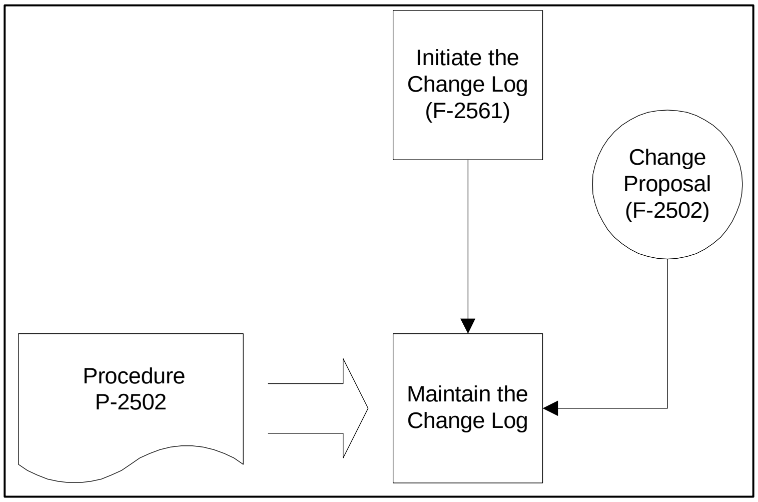


Figure 9–2. Maintain the Change Log

Activity 4.2: Engineer the Detailed Design

See Figure 9–3. Given the scope of the work to be done as documented in the SRS, design documents, Project Charter, the previous analysis of alternatives, and final System Operations Concept, select an engineering approach that both maximizes reuse and the probability of success.

Perform the following sub-activities:

1. Determine if there is an off-the-shelf solution that can be directly installed
2. If not, determine if there is an off-the-shelf solution that can be modified and installed.
3. If not, plan to develop the system from scratch.
4. Using a top-down approach, partition the system into modules, specifying which parts can be reused or modified or built from scratch.
5. Review the life-cycle approach selected and schedule the development and integration of the modules and document this in the Project Management Plan and Software Development Plan.

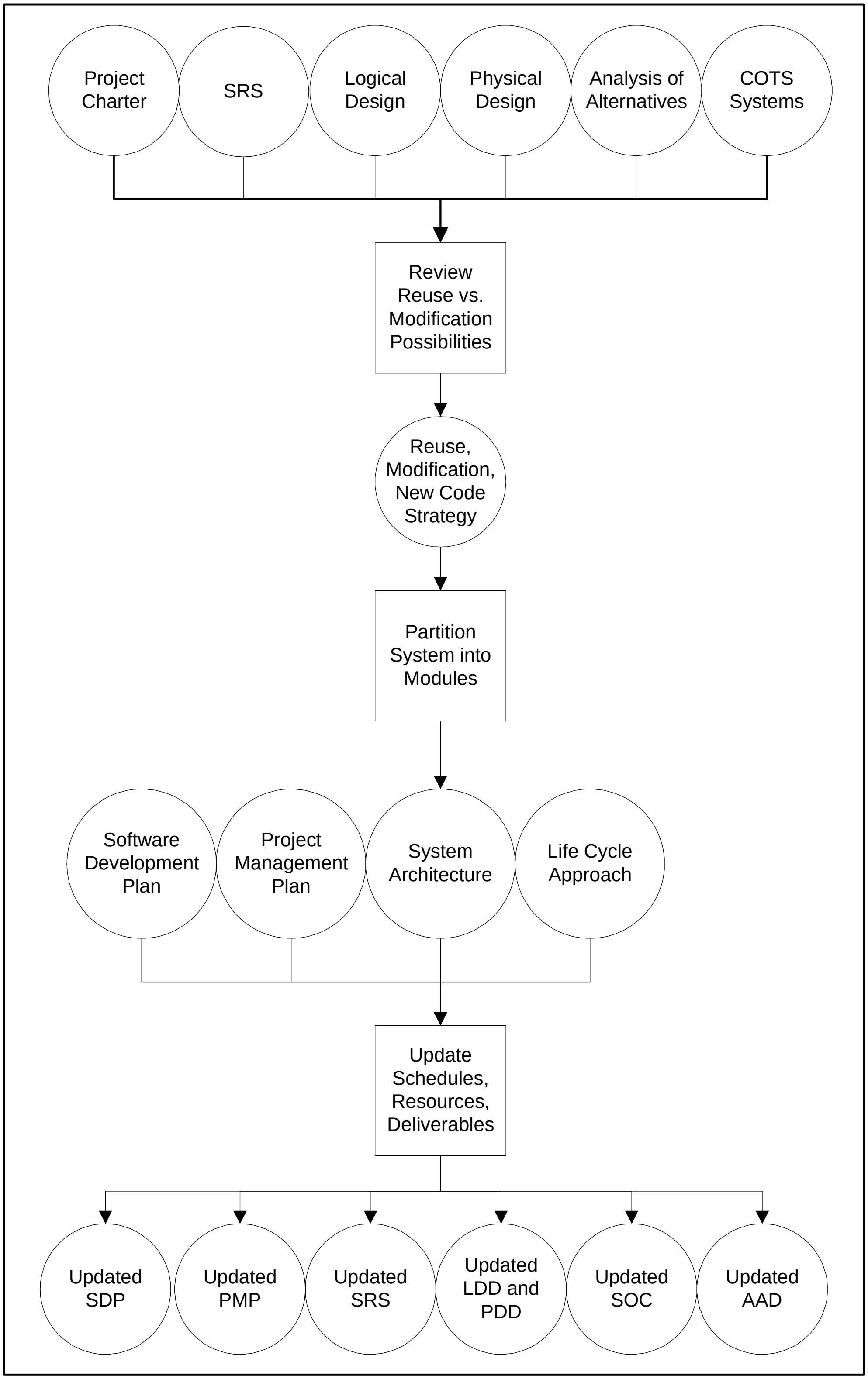


Figure 9–3. Engineer the Detailed Design

Activity 4.3: Build the Solution

See Figure 9–4, Using the design materials produced as a part of Component 3 and earlier activities within Component 4:

1. Build database structure and populate the database
2. Develop program code
3. Develop external systems interfaces and update design and integration documentation, as required.

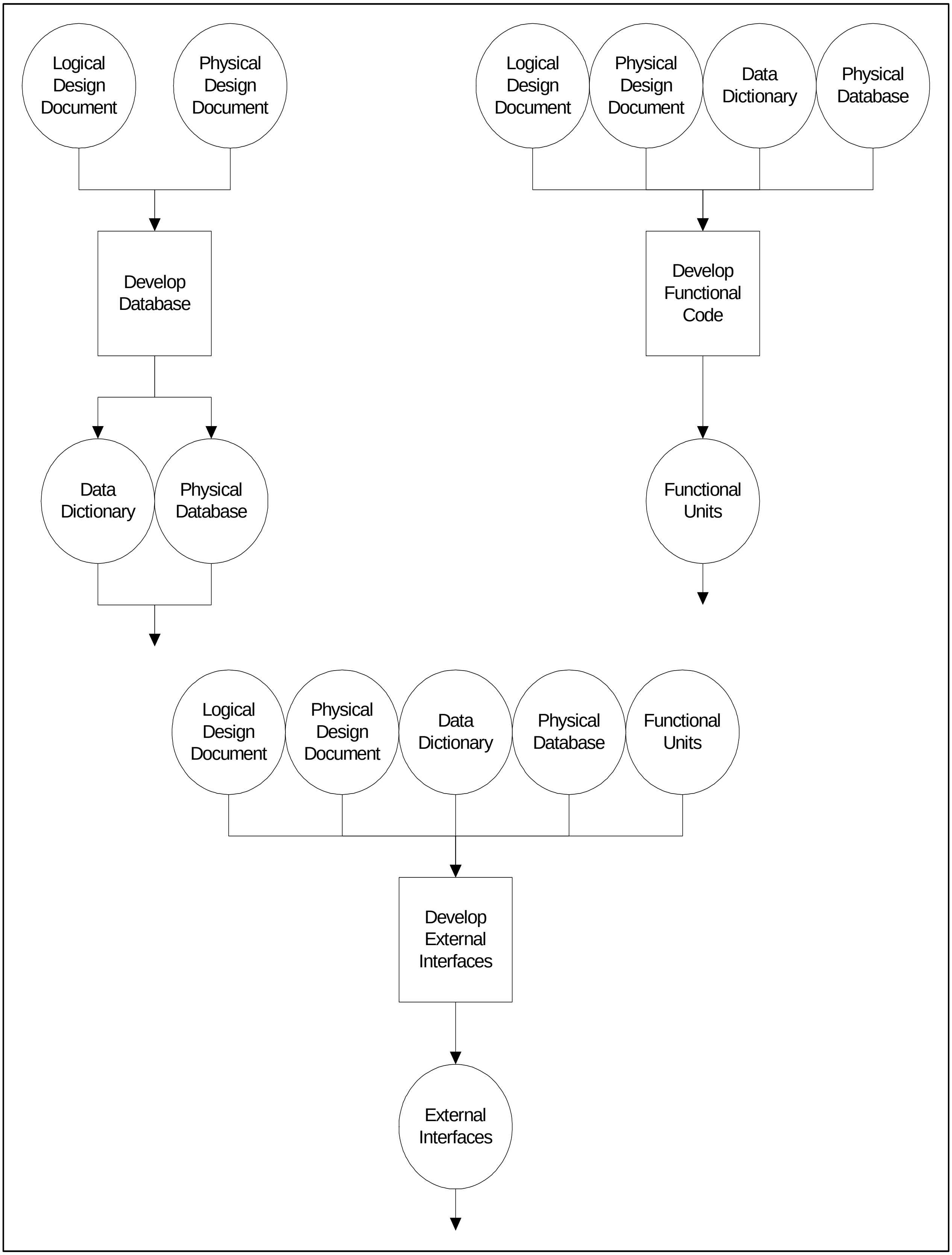


Figure 9–4. Build the Solution

Activity 4.4: Integrate the Solution

See Figure 9–5. As parts of a solution are built, they must be integrated into larger and larger parts until the system is built and integrated into the overall enterprise. Integrate the system in a step-wise fashion as follows:

1. Integrate units into modules.
2. Integrate modules into systems.
3. Integrate the systems into the network of systems, that is, the enterprise.

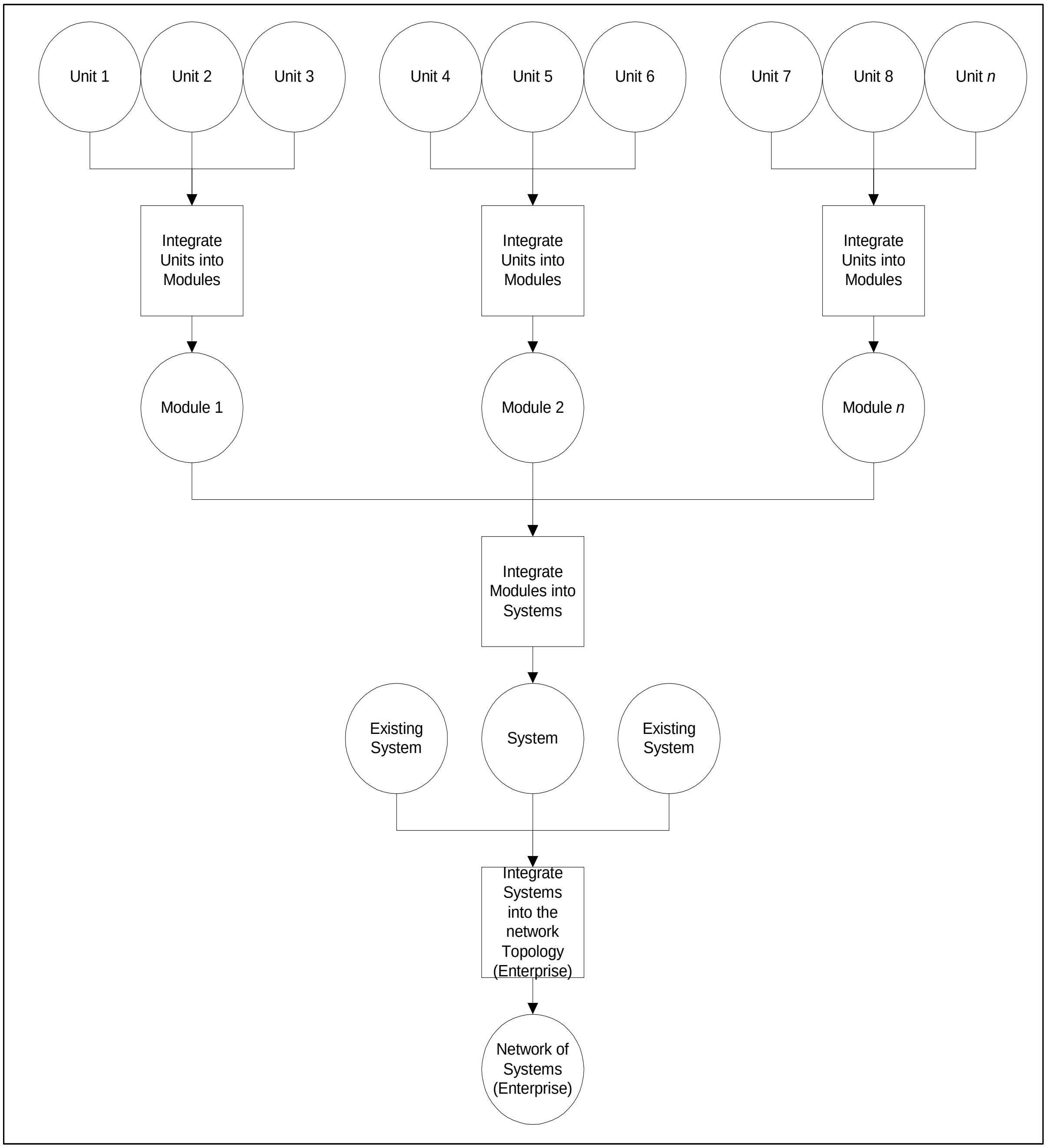

Unit 1
Unit 2
Unit 3
Unit 4
Unit 5
Unit 6
Unit 7
Unit 8
Unit *n*
Integrate Units into Modules
Integrate Units into Modules
Integrate Units into Modules
Module 1
Module 2
Module *n*
Integrate Modules into Systems
Existing System
System
Existing System
Integrate Systems into the network Topology (Enterprise)
Network of Systems (Enterprise)

Figure 9–5. Integrate the Solution

Activity 4.5: Test the Solution

See Figure 9–6. As each part of the solution is built, it should be tested before being integrated into a larger part that is also tested. Just as the Build the Solution activity of the Engineer the Solution Component is bottom up, so is the testing activity.

Note:
It is recognized that Unit, Module, and System/Acceptance testing integrate software elements at different levels. Figure 9-6 expresses all integrated elements as "Component."

For each kind of test:

- Unit
- Module
- System
- Acceptance

Perform each of the following sub-activities:

- Develop test data. (This requires participation by the Business Advocates and Overall Project Manager)
- Develop a test plan (that is, a test scenario or execution plan that specifies the inputs and activities/processes required for performing each test).
    - Development of test scenarios and execution plans for User testing requires the participation of user Business Advocates, Overall Project Manager, etc. ONLY Prioritized Requirements that are defined as Critical or Would Improve the Process and/or Provide Ease of Use to the User will be tested against the corresponding scenarios for Acceptance Testing.
    - Development of unique test scenarios and execution plans for integration testing (throughput, cohabitation, etc.) requires the participation of the Infrastructure representative(s).
    - Development of test scenarios and execution plans for business cases and business rules testing requires the participation of the Business Advocates and Overall Project Manager.
- Document the expected results.
- Execute the test and record actual results.
- Analyze the test results. Compare the actual results with the expected results.
- If the test results agree with the anticipated results, then enter the component into the Project/System CM library.
- If test results do not agree with anticipated results, generate problem report(s) (if applicable), investigate the cause, and generate change requests to fix the problem.

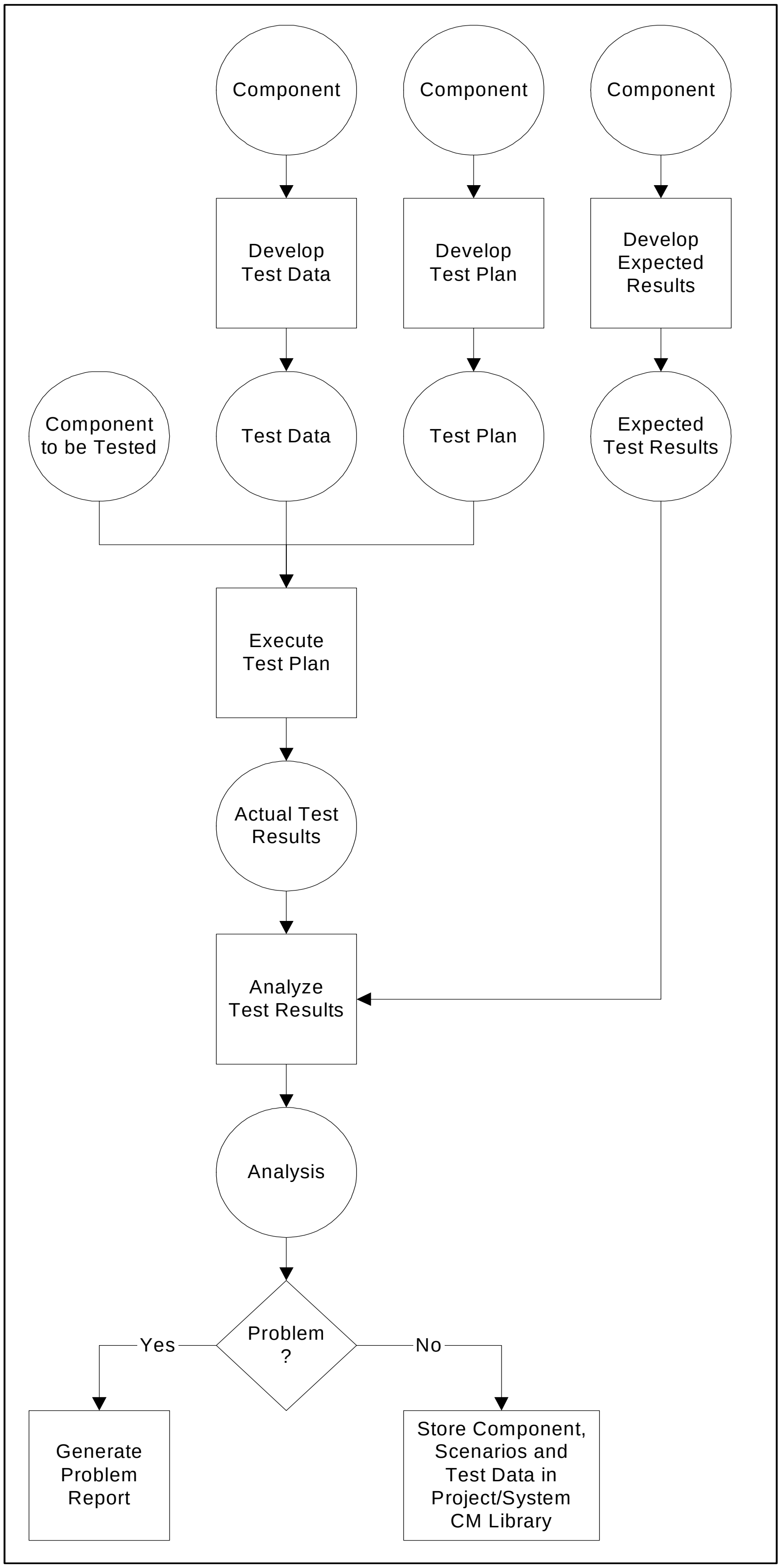


Figure 9–6. Test the Solution

Activity 4.6: Create the Rollout Strategy and Continue Deployment Planning

See Figure 9–7. Given the locations and organizations that will receive the new information management system, develop a rollout strategy and document this in the Tactical Integration Plan. If appropriate, entries may be made to the SEN.

1. Define the target users
2. Survey the target users desktops to determine their capabilities and needs for upgrades (Target Users Capability Upgrade Request Form, F-6054)
3. Create a schedule for the upgrades
4. Create instruction and a schedule for the installation of the new system
5. Package the Rollout Strategy as part of the Tactical Integration Plan.

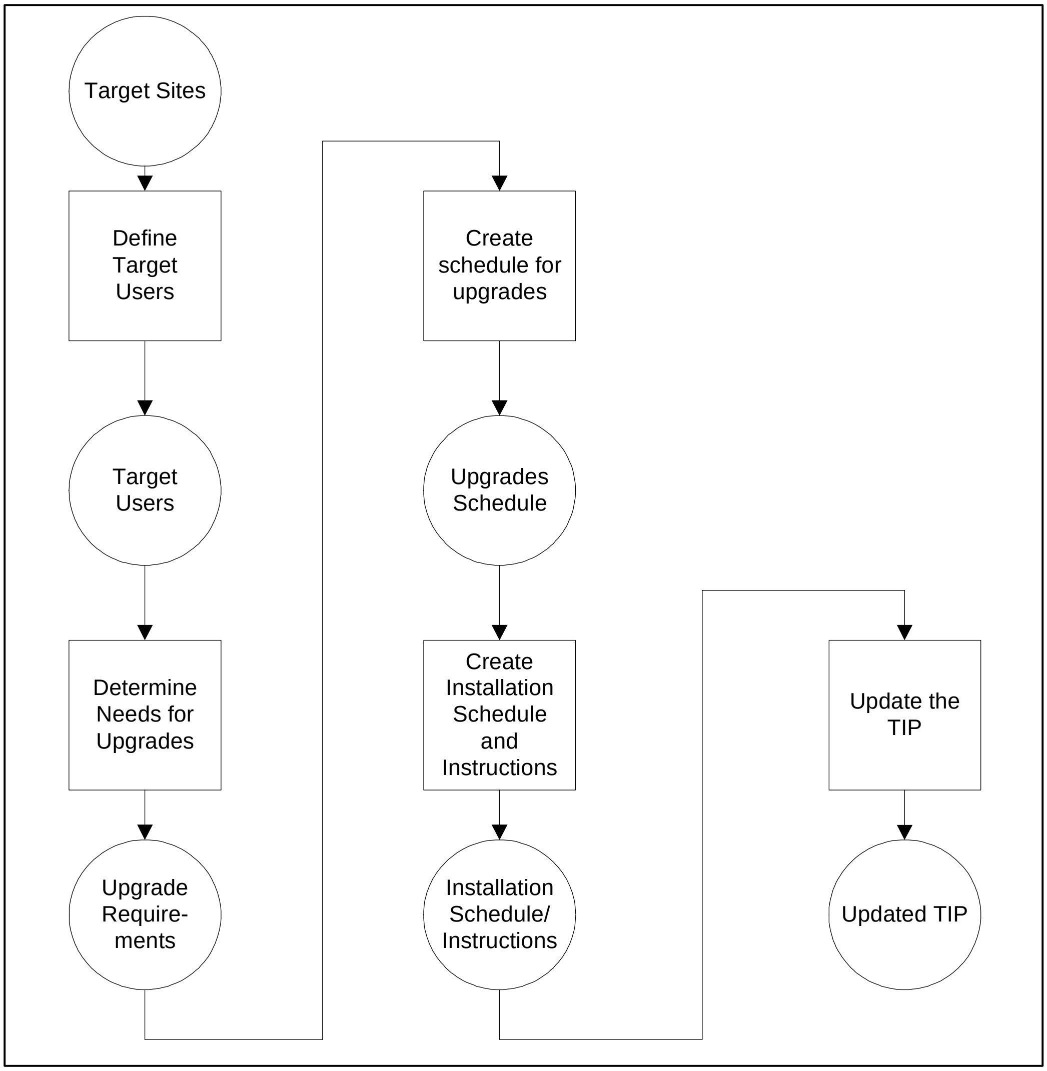

Target Sites
Define Target Users
Target Users
Determine Needs for Upgrades
Upgrade Require-ments
Create schedule for upgrades
Upgrades Schedule
Create Installation Schedule and Instructions
Installation Schedule/ Instructions
Update the TIP
Updated TIP

Figure 9–7. Create the Rollout Strategy and Continue Deployment Planning

Activity 4.7: Update Policies and Procedures

See Figure 9–8. Update appropriate policies and procedures (for example, production schedules, business guidance (MD, Rule Change, etc.) or support).

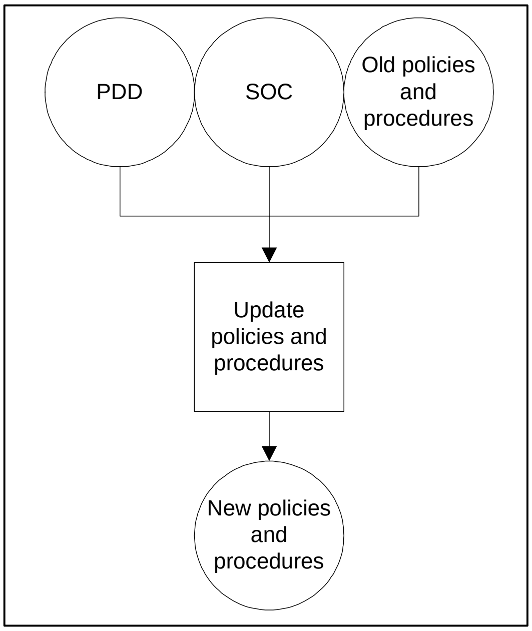

PDD
SOC
Old policies and procedures
Update policies and procedures
New policies and procedures

Figure 9–8. Update Policies and Procedures

Activity 4-8: Build the User Manual and Training Materials

See Figure 9–9. Using (a) the plan for building an integrated set of education, training, and support materials from the Design the Solution component, (b) the design, and (c) the actual system specifications, build the materials to support the users.

- Produce user guides
- Produce a help system
- Produce operational guides
- Produce tutorials
- Create and maintain a customer support center
- Develop a user training plan

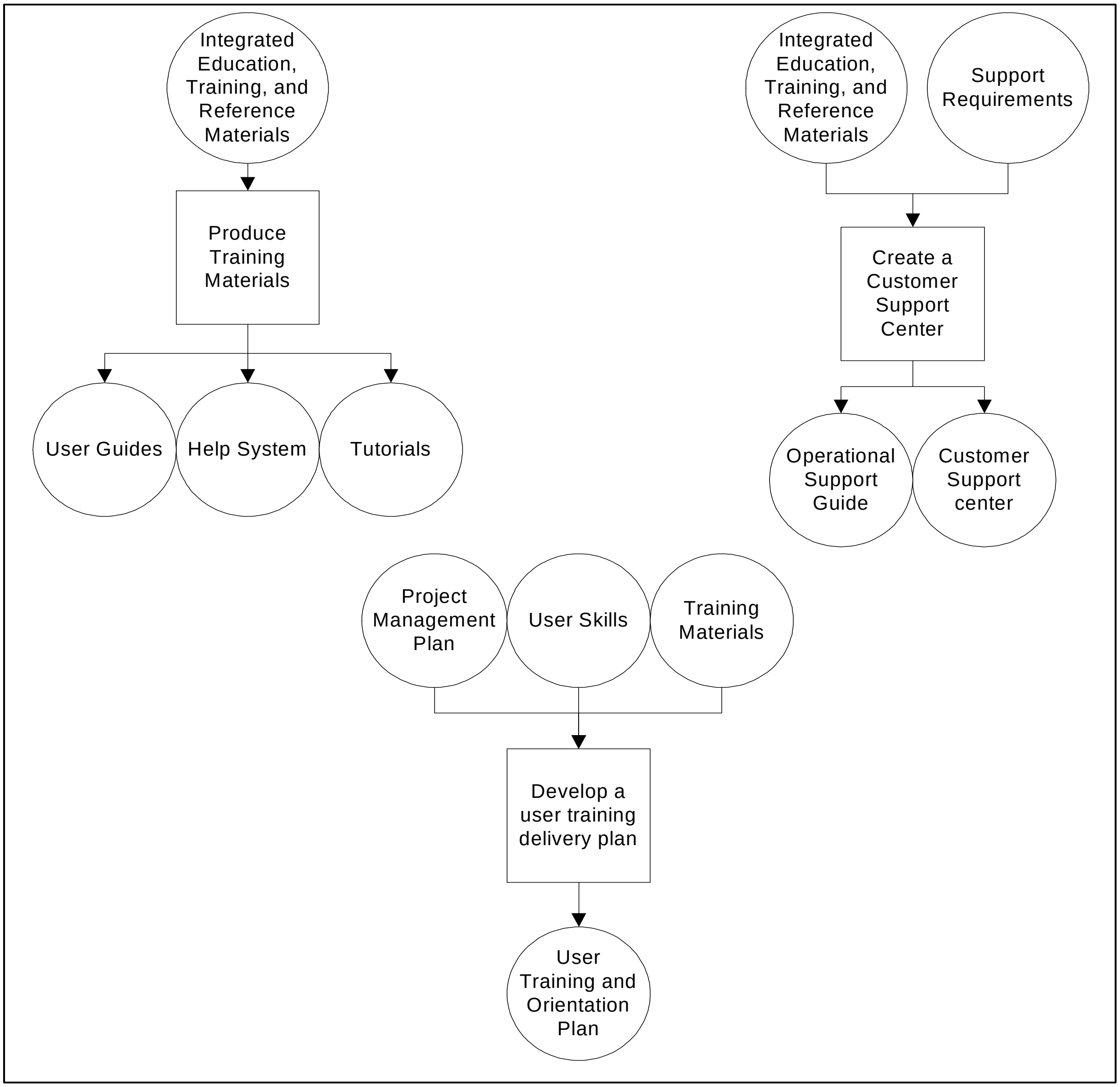

Integrated Education, Training, and Reference Materials
Produce Training Materials
User Guides
Help System
Tutorials
Integrated Education, Training, and Reference Materials
Support Requirements
Create a Customer Support Center
Operational Support Guide
Customer Support center
Project Management Plan
User Skills
Training Materials
Develop a user training delivery plan
User Training and Orientation Plan

Figure 9–9. Build the User Manual and Training Materials

Activity 4.9: Deliver the Training

See Figure 9–10. Deliver appropriate training, as planned, to all affected personnel.

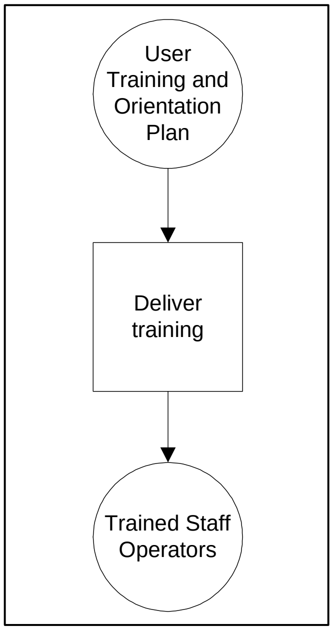


Figure 9–10. Deliver Training

Activity 4.10: Update the Project Management Plan and Software Development Plan

See Figure 9–11. Prepare for the next component by updating the Project Management Plan with information produced during this component.

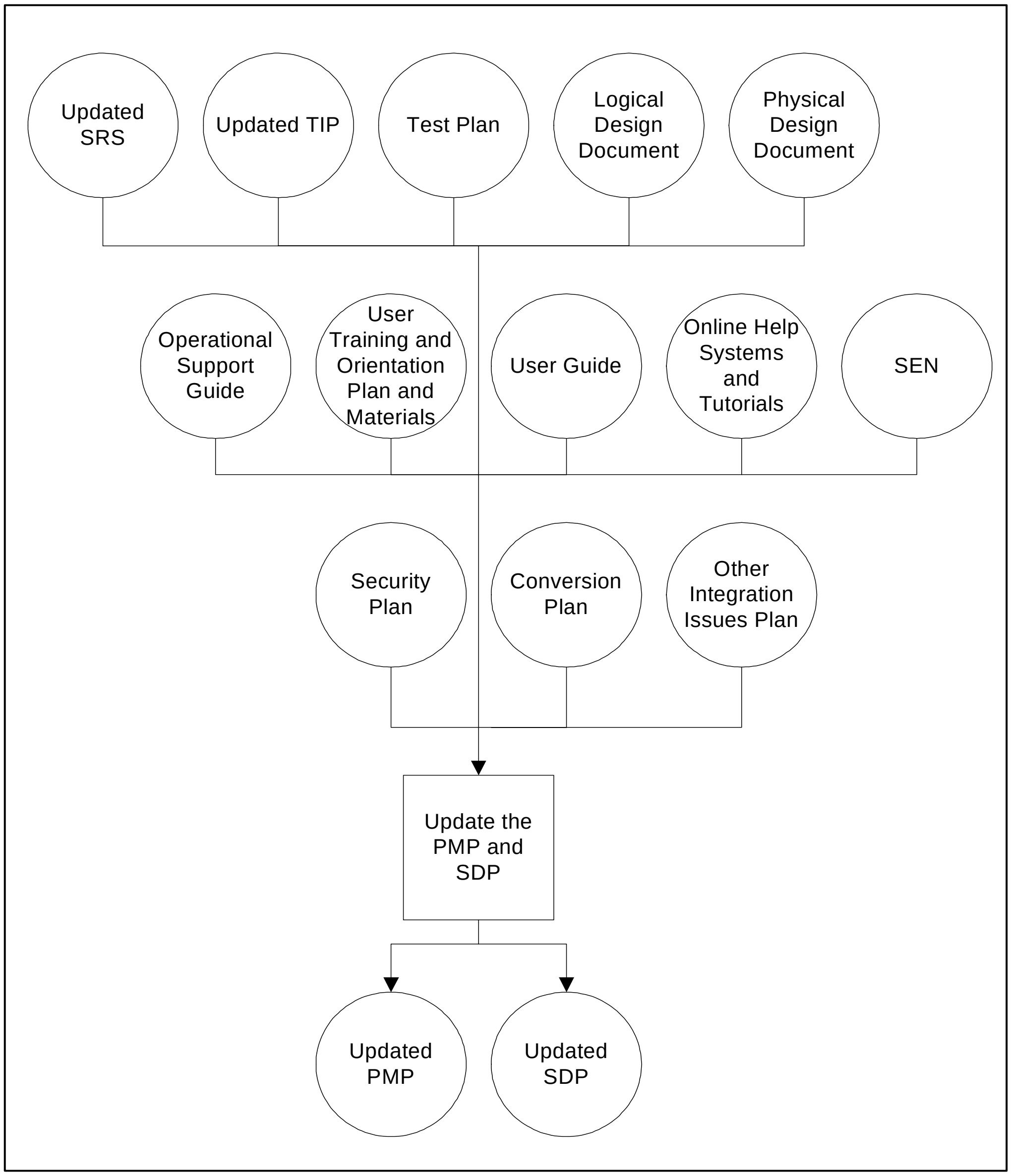


Figure 9–11. Update the Project Management Plan

Activity 4.11: Review (Component 4) and Transfer Materials to Central Configuration Management

See Figure 9–12. Review the results of the activities that were performed in this component to determine if the system should be deployed. Include the Records Management organization in this review activity. After review and approval, transfer all Project-level products and materials to Central Configuration Management (CCM)

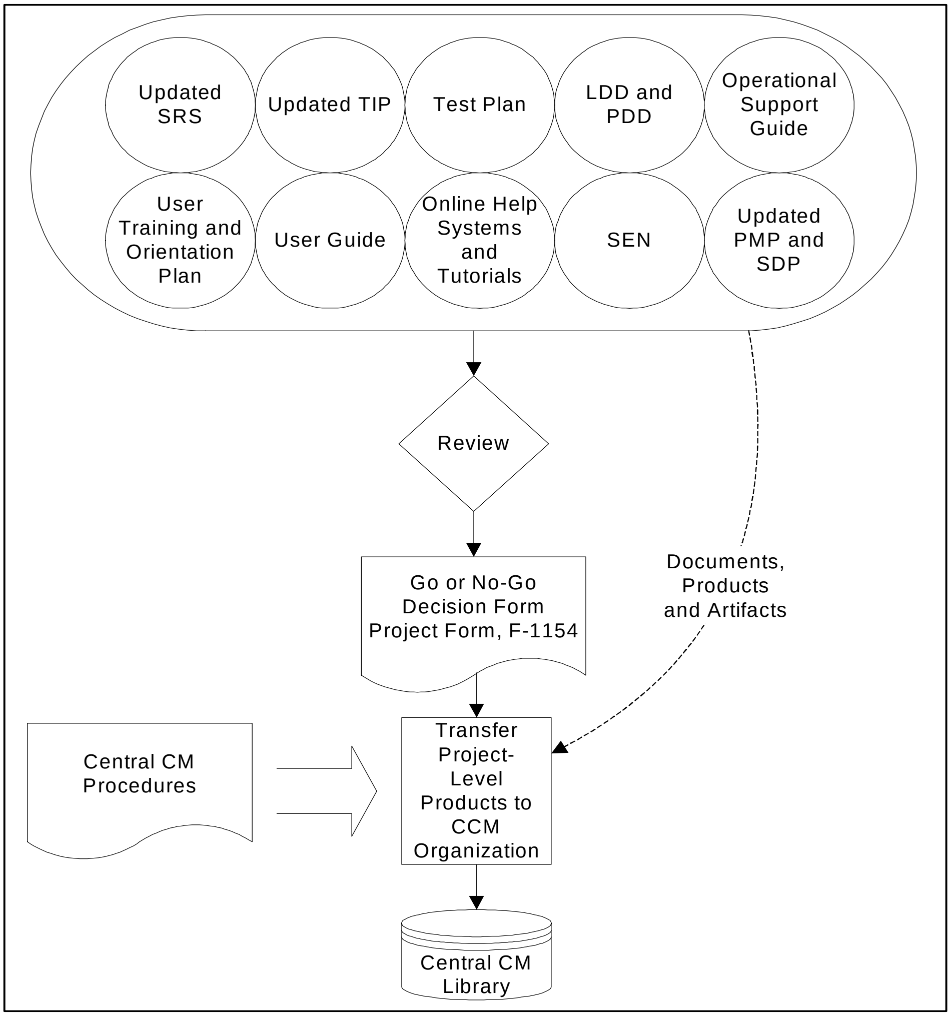

Updated SRS
Updated TIP
Test Plan
LDD and PDD
Operational Support Guide
User Training and Orientation Plan
User Guide
Online Help Systems and Tutorials
SEN
Updated PMP and SDP
Review
Go or No-Go Decision Form Project Form, F-1154
Documents, Products and Artifacts
Central CM Procedures
Transfer Project-Level Products to CCM Organization
Central CM Library

Figure 9–12. Review (Component 4)

10. Component 5. Deploy the Solution

10.1 Purpose

The purpose of this component is to roll out the integrated solution, which includes:

- Review and update deployment plans and announce deployment
- Validate the environment and update it if required
- Install the solution in a full test environment (modeled to production)
- Test the installed solution (full integration and operational)
- Implement Business Policies and Procedures
- Training the users
- User and Acceptance test of the system
- Review the results of the deployment and test
- Prepare feedback reports for all testing
- Obtain user approval
- Update the Production Environment
- Deploy/Install the new/updated system
- Cut over to a production environment
- Use the system
- Update the technical inventory

10.2 Roles and Responsibilities

The roles required to perform the activities in the Deploy the Solution component are shown in Table 10–1.

Table 10–1. Component 5 Roles and Responsibilities

Roles	Responsibilities
Overall Project Manager Technical Project Manager Development Team	Accountable for Execution of Activities
Business Advocate	Approval to Proceed

Roles	Responsibilities
Executive Sponsor Business project Manager Business SME Technical SME Other Representatives or Stakeholders • Quality Assurance Manager • Configuration management Manager	Provides Input

10.3 Entry Criteria

The following set of events trigger the initiation of Component 5:

- System test completed
- Rollout plan completed
- Training plan and materials completed
- Installation instructions completed
- Regression test completed
- Customer acceptance

10.4 Input, Activities, and Outputs

The inputs to this component are as follows:

- Project Management Plan
- Tactical Integration Plan, including
  - Conversion Plan
  - Rollout Plan
  - User Training and Orientation Plan and materials
  - Installation instructions
  - Performance criteria measures

Figure 10–1 illustrates the flow of the following activities:

1. Review and Update Plans and Announce Deployment
2. Validate and Upgrade the Operational Test Environment
3. Install the Solution
4. Test the Installed Solution
5. Analyze the Installed System Test Results
6. Test the Operational Support System
7. Analyze the Operational Support Test Results

8. Implement Policies and Procedures
9. Train the Users
10. User Test the Solution
11. Analyze the User Test Results
12. Acceptance Test the Solution
13. Analyze the Acceptance Test Results
14. Validate and Upgrade the Production
15. Deploy the Solution to Production Environment
16. Review (Component 5)

The inputs to this component were produced as products of Component 4.

As the figure suggests, many of the activities can be performed in parallel. The activities are described in more detail in Section 3.10.

Figure 10–1 also summarizes the outputs of all Component 5 activities in the central data store.

Appendix C includes a summary description of all products of NRC projects, indicates within which components each product is created and updated, and specifies which products are required.
The products, whose standard and form numbers are shown in the figure, are described in more detail in the companion volume *SDLCM Methodology Procedures, Standards, and Forms*.

10.5 Techniques and Tools

The activities of Component 5 are supported by the following techniques.

- Structured Facilitation
- Structured Walkthrough
- Peer Review

Refer to the current *SDLCM Methodology Tool Inventory* for the recommended set of tools.

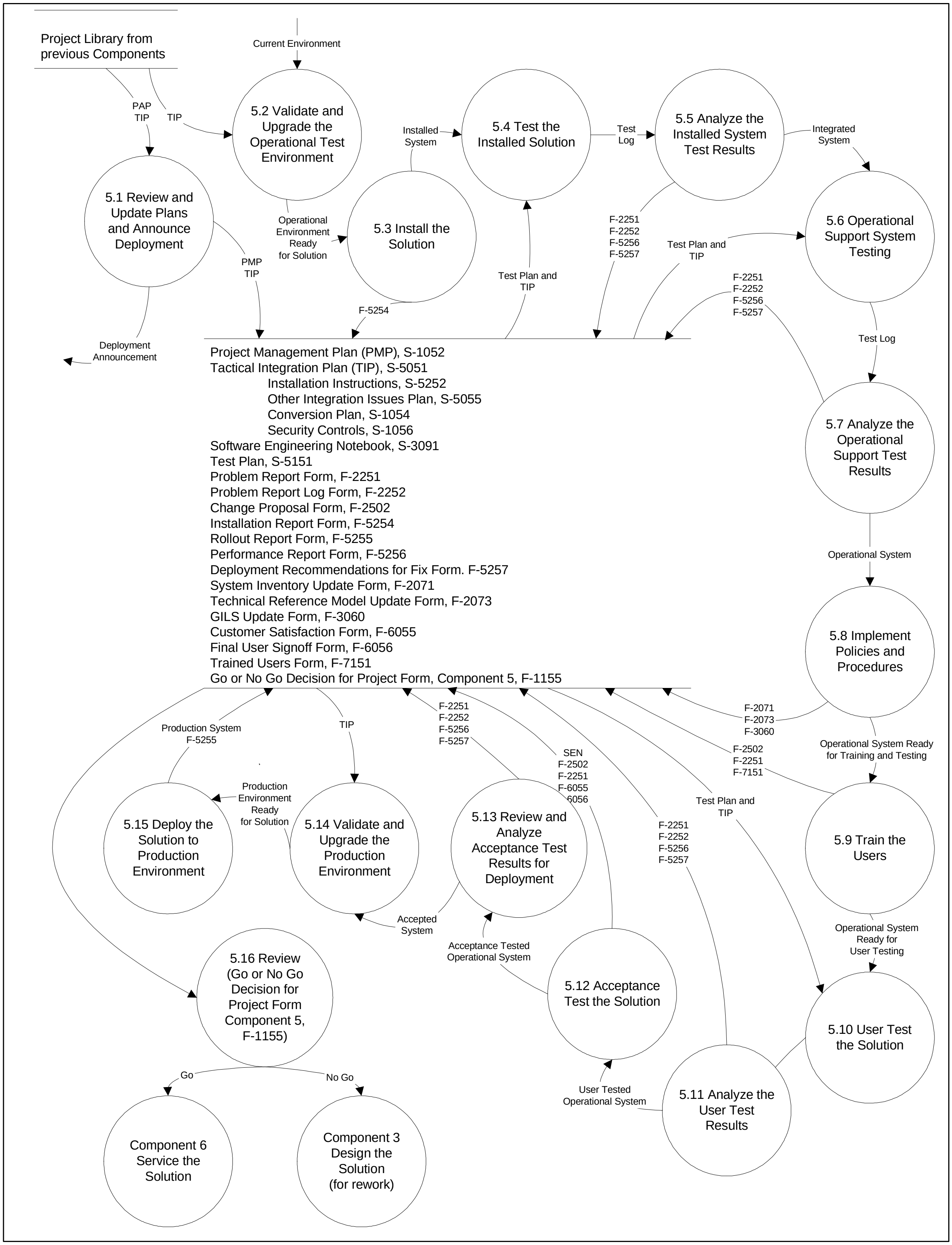

Project Library from previous Components
Current Environment
PAP
TIP
TIP
5.1 Review and Update Plans and Announce Deployment
5.2 Validate and Upgrade the Operational Test Environment
Operational Environment Ready for Solution
5.3 Install the Solution
Installed System
5.4 Test the Installed Solution
Test Log
5.5 Analyze the Installed System Test Results
Integrated System
5.6 Operational Support System Testing
PMP
TIP
F-5254
Test Plan and TIP
F-2251
F-2252
F-5256
F-5257
Test Plan and TIP
F-2251
F-2252
F-5256
F-5257
Test Log
Deployment Announcement
Project Management Plan (PMP), S-1052
Tactical Integration Plan (TIP), S-5051
Installation Instructions, S-5252
Other Integration Issues Plan, S-5055
Conversion Plan, S-1054
Security Controls, S-1056
Software Engineering Notebook, S-3091
Test Plan, S-5151
Problem Report Form, F-2251
Problem Report Log Form, F-2252
Change Proposal Form, F-2502
Installation Report Form, F-5254
Rollout Report Form, F-5255
Performance Report Form, F-5256
Deployment Recommendations for Fix Form. F-5257
System Inventory Update Form, F-2071
Technical Reference Model Update Form, F-2073
GILS Update Form, F-3060
Customer Satisfaction Form, F-6055
Final User Signoff Form, F-6056
Trained Users Form, F-7151
Go or No Go Decision for Project Form, Component 5, F-1155
5.7 Analyze the Operational Support Test Results
Operational System
5.8 Implement Policies and Procedures
F-2071
F-2073
F-3060
Operational System Ready for Training and Testing
F-2502
F-2251
F-7151
5.9 Train the Users
Operational System Ready for User Testing
5.10 User Test the Solution
Test Plan and TIP
F-2251
F-2252
F-5256
F-5257
5.11 Analyze the User Test Results
User Tested Operational System
5.12 Acceptance Test the Solution
Acceptance Tested Operational System
5.13 Review and Analyze Acceptance Test Results for Deployment
SEN
F-2502
F-2251
F-6055
F-6056
F-2251
F-2252
F-5256
F-5257
Accepted System
TIP
5.14 Validate and Upgrade the Production Environment
Production Environment Ready for Solution
5.15 Deploy the Solution to Production Environment
Production System
F-5255
5.16 Review (Go or No Go Decision for Project Form Component 5, F-1155)
Go
No Go
Component 6 Service the Solution
Component 3 Design the Solution (for rework)

Figure 10–1. Component 5

10.6 Exit Criteria

Component 5 Is complete when:

- All critical products have been reviewed (structured walkthrough or peer review)
- Key products have been inspected by QA to conform to project standards
- Products have been placed under CM control
- Information has been shared
- Issues have been resolved
- There is management buy-in to move forward
- There are no reasons to stop this project now

10.7 Component 5 Activity Details

The following pages provide detailed activities of Component 5, Deploy the Solution.

Activity 5.1: Review and Update Plans and Announce Deployment

See Figure 10–2. Review the Project Management Plan and TIP, and announce the deployment. It is important to prepare people to expect deployment activities.

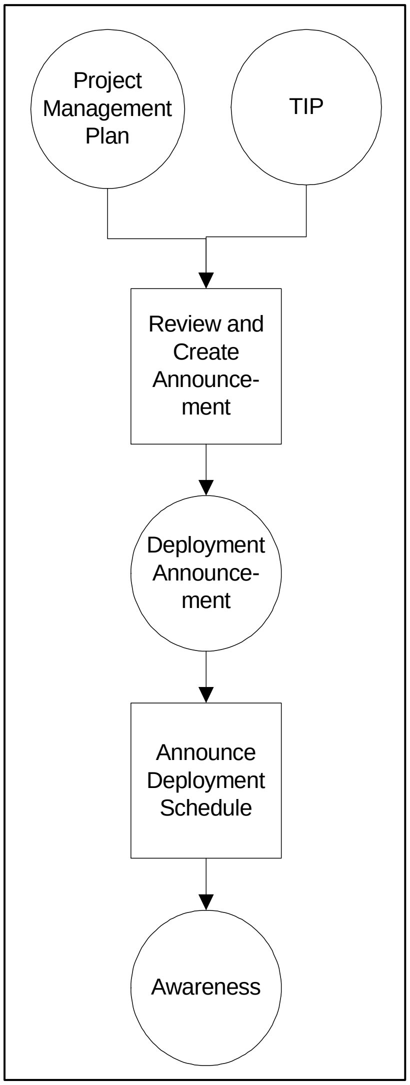


Figure 10–2. Review and Update Plans and Announce Deployment

Activity 5.2: Validate and Upgrade the Operational Test Environment

See Figure 10–3. Using the TIP:

1. Validate the operational test environment (workstation and host or client and server) to determine what the system needs to run and whether the system will run in each individual user environment.
2. If needed, procure and install any additional hardware and software.
3. Upgrade the operational test environment.

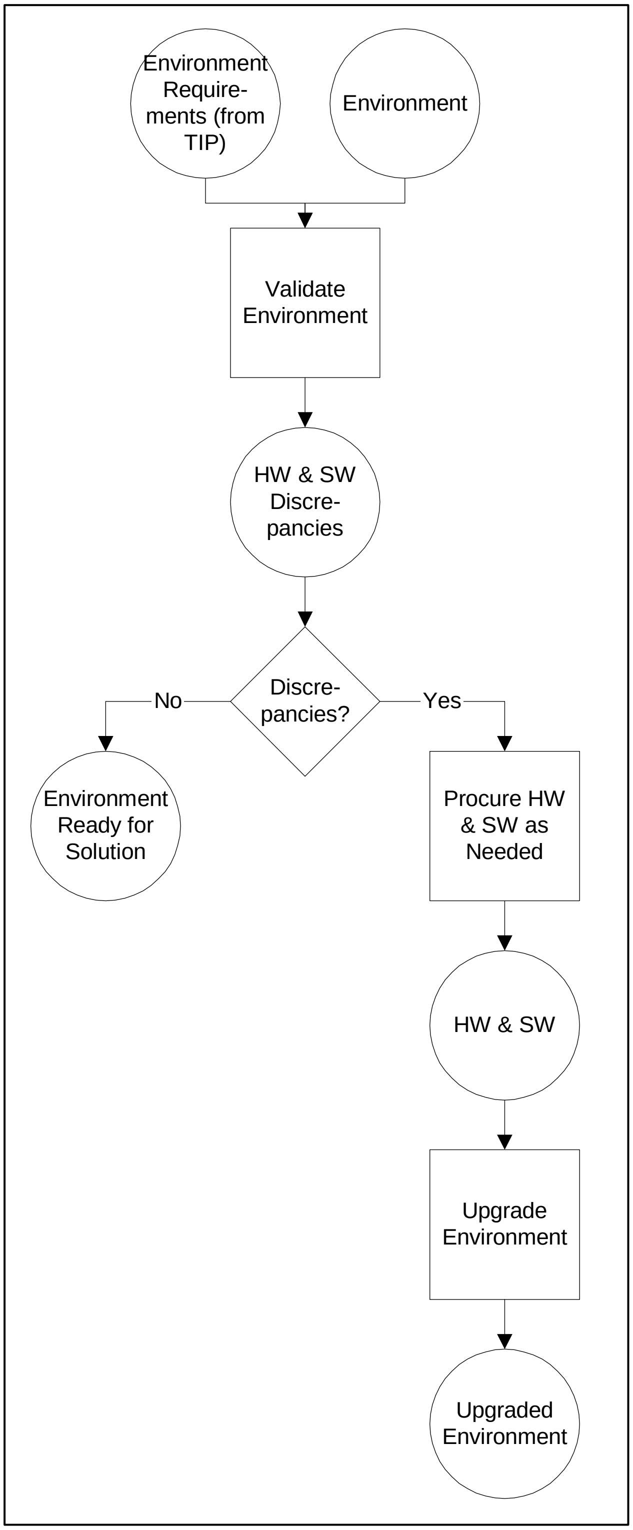


Figure 10–3. Validate and Upgrade the Environment

Activity 5.3: Install the Solution

See Figure 10–4. After the operational test environments are validated and updated:

1. Install the solution, including the solution modules and database, using the installation instructions.
2. Capture the results of the installation in an installation report (Form F–5254).

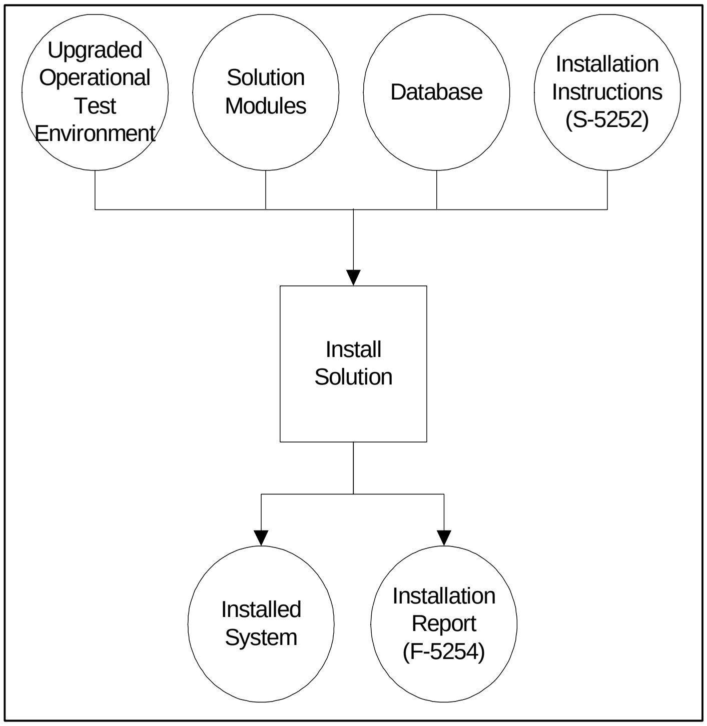

Upgraded Operational Test Environment
Solution Modules
Database
Installation Instructions (S-5252)
Install Solution
Installed System
Installation Report (F-5254)

Figure 10–4. Install the Solution

Activity 5.4: Test the Installed Solution

See Figure 10–5. After the Application System is installed:

1. Test it against performance specifications and functional requirements.
2. If there are any discrepancies, log them in a problem report log, and resolve the issues in accordance with project procedures.

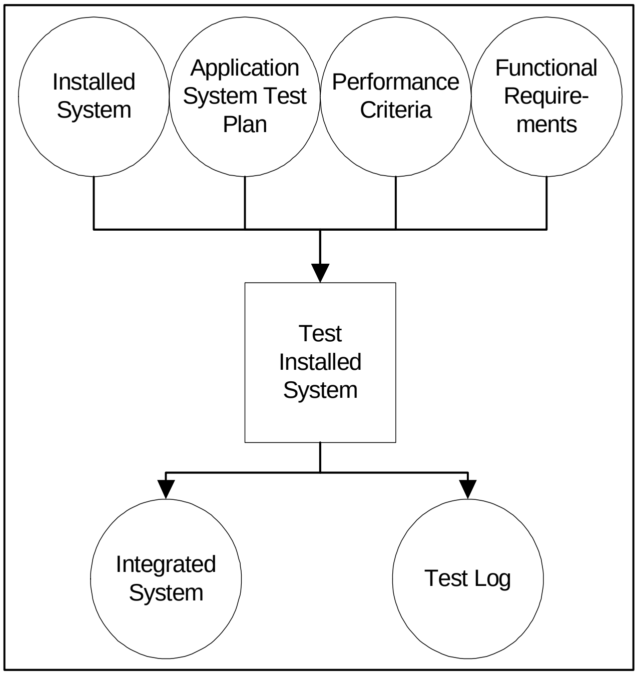


Figure 10–5. Test the Installed Solution

Activity 5.5: Analyze the Installed System Test Results

See Figure 10–6. Using the results of the Application System test:

1. Review the test results. Log your analysis in a feedback report.
2. If there are discrepancies, log them as problem reports (Forms F–2251 and F–2252) to be resolved in accordance with proper procedures.
3. If there are no discrepancies, then the system is ready for operational support testing.

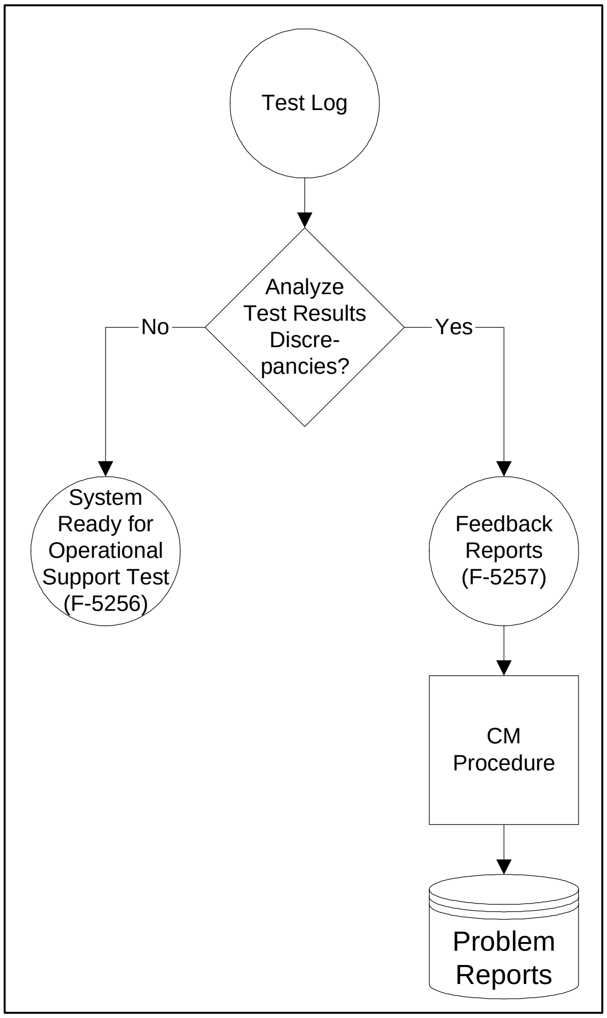

Test Log
Analyze Test Results Discre- pancies?
No
Yes
System Ready for Operational Support Test (F-5256)
Feedback Reports (F-5257)
CM Procedure
Problem Reports

Figure 10–6. Analyze the Test Results

Activity 5.6: Test the Operational Support System

See Figure 10–7. After the Application System has completed System Integration Test:

1. Test the Integrated System against performance specifications and functional requirements viewed from the operational support perspective.
2. If there are any discrepancies, log them in a problem report log, and resolve the issues in accordance with project procedures.

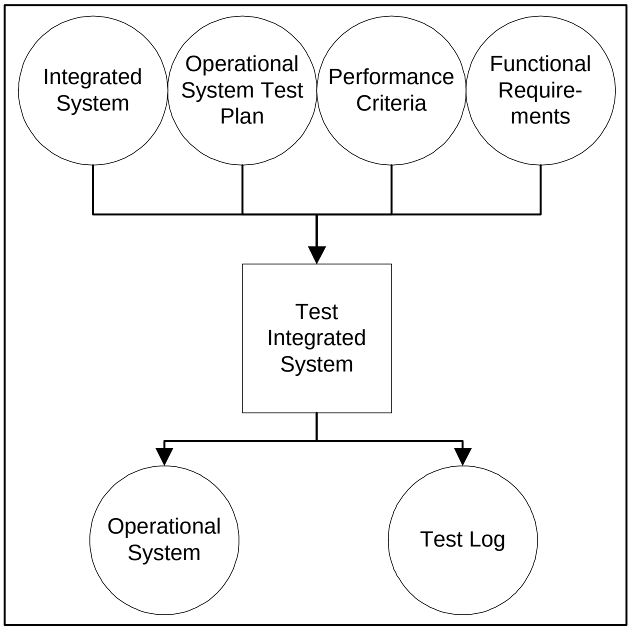


Figure 10–7. Test the Integrated System

Activity 5.7: Analyze the Operational Support Test Results

See Figure 10–8. Using the results of the system test:

1. Review the test results. Log your analysis in a feedback report.
2. If there are discrepancies, log them as problem reports (Forms F–2251 and F–2252) to be resolved in accordance with proper procedures.
3. If there are no discrepancies, then the system is ready to be used to train the users.

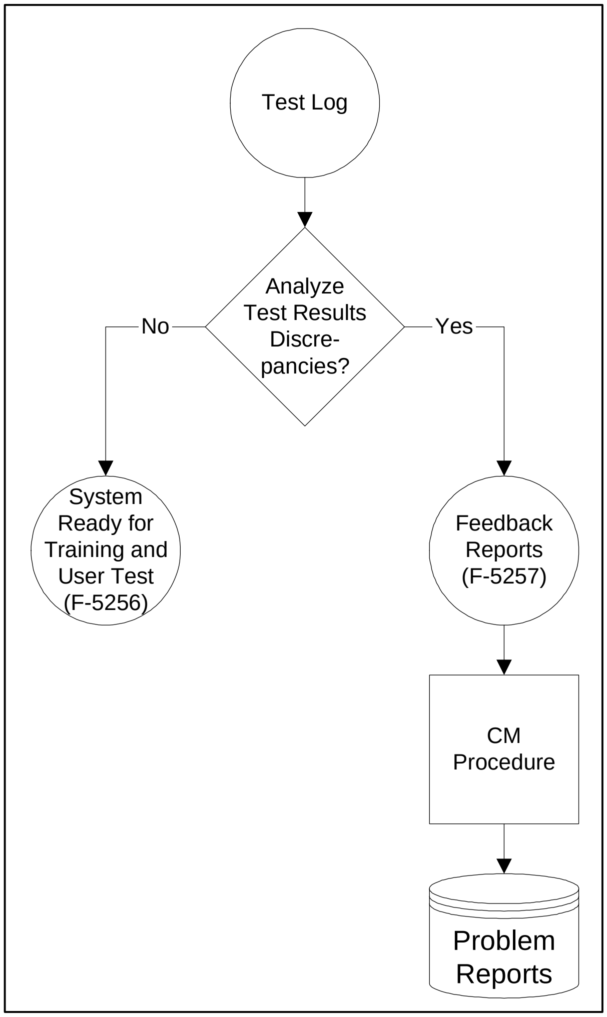

Test Log
Analyze Test Results Discre-pancies?
No
Yes
System Ready for Training and User Test (F-5256)
Feedback Reports (F-5257)
CM Procedure
Problem Reports

Figure 10–8. Analyze the Test Results

Activity 5.8: Implement Policies and Procedures

See Figure 10–9. After the system has been installed, tested, and in the operational state, implement the policies and procedures (for example, production schedules or support) that were updated in Component 4, Activity 4.8.

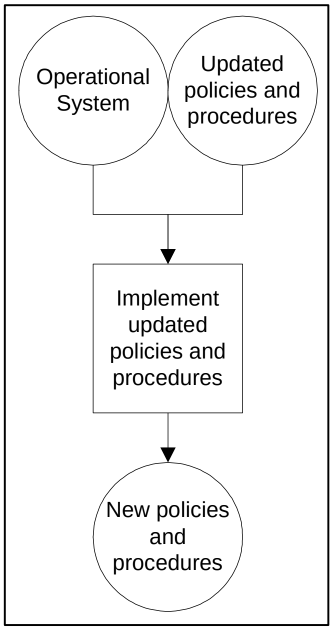

Operational System
Updated policies and procedures
Implement updated policies and procedures
New policies and procedures

Figure 10–9. Implement Policies and Procedures

Activity 5.9: Train the Users

See Figure 10–10. Using the training plan and training materials prepared during the Engineer the Solution component:

1. Train the users (F–7151).
2. Generate any needed problem reports (F–2251) or change proposals (F–2502).

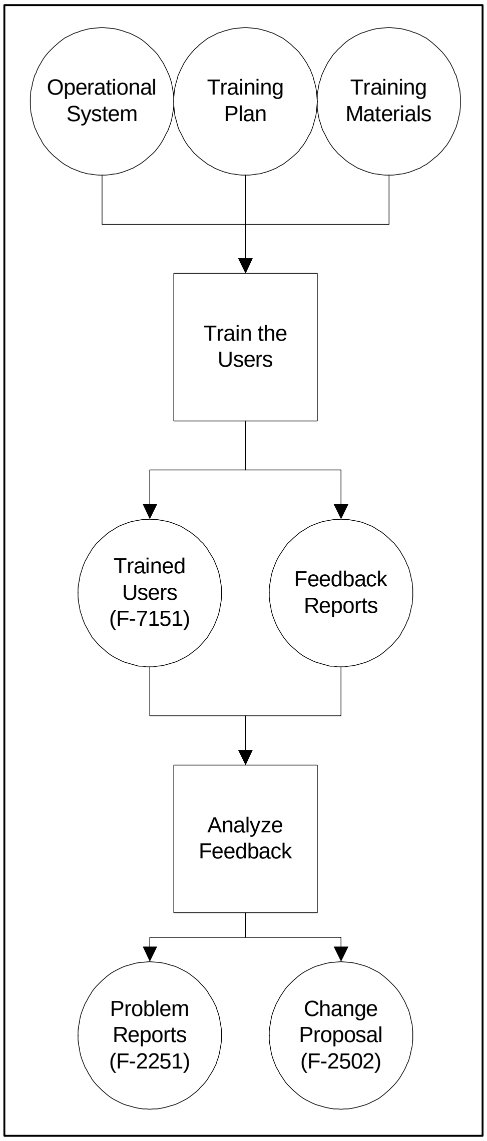


Figure 10–10. Train the Users

Activity 5.10: User Test the Solution

See Figure 10–11. After the Operational System has completed Operational Support System Integration Test:

1. Test the Operational System against performance specifications and functional requirements viewed from the users' perspective.
2. If there are any discrepancies, log them in a problem report log, and resolve the issues in accordance with project procedures.

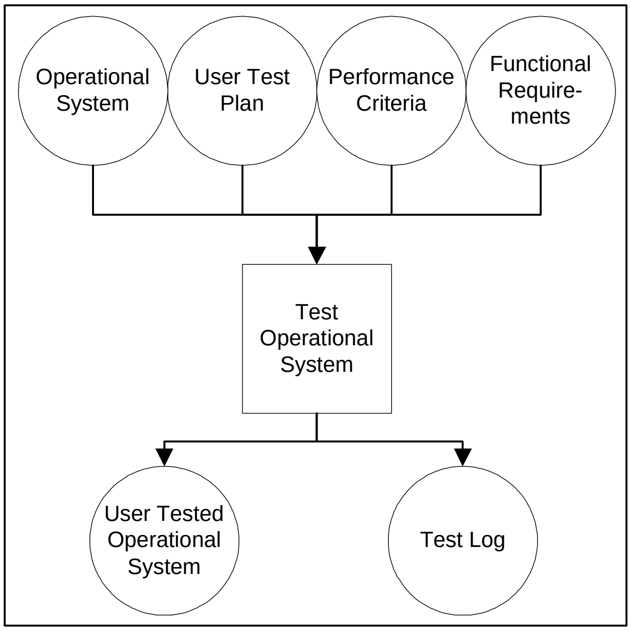


Figure 10–11. User Test the Solution

Activity 5.11: Analyze the User Test Results

See Figure 10–12. Using the results of the user test of the system:

1. Review the test results. Log your analysis in a feedback report.
2. If there are discrepancies, log them as problem reports (Forms F–2251 and F–2252) to be resolved in accordance with proper procedures.
3. If there are no discrepancies, then the system is ready to be acceptance tested by the customer.

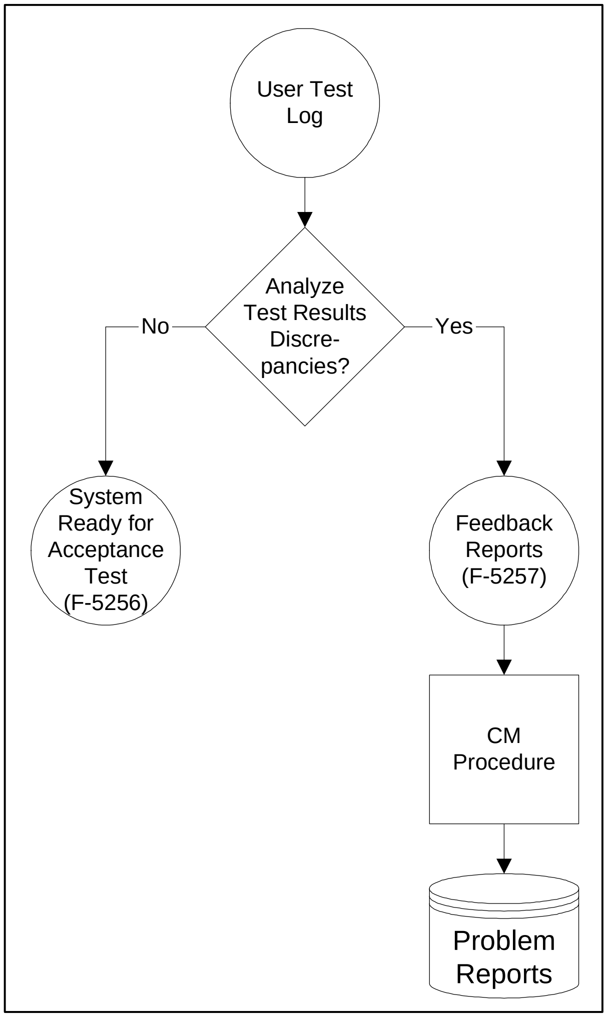

User Test Log
Analyze Test Results Discre-pancies?
No
Yes
System Ready for Acceptance Test (F-5256)
Feedback Reports (F-5257)
CM Procedure
Problem Reports

Figure 10–12. Analyze the User Test Results

Activity 5.12: Acceptance Test the Solution

See Figure 10–13. After the system has been installed, has completed Integration and Operational Support testing, all known bugs have been eliminated, a cadre of users has been trained, and users' testing is complete, then the system is ready for customer acceptance testing:

1. Obtain user approval for technical environment, performance against criteria, cohabitation (test system performance).
2. Collect customer satisfaction measures (F–6055).
3. Conduct review and process any new discrepancies (F–2251 or F–2502).
4. Obtain final user sign-off (F–6056).

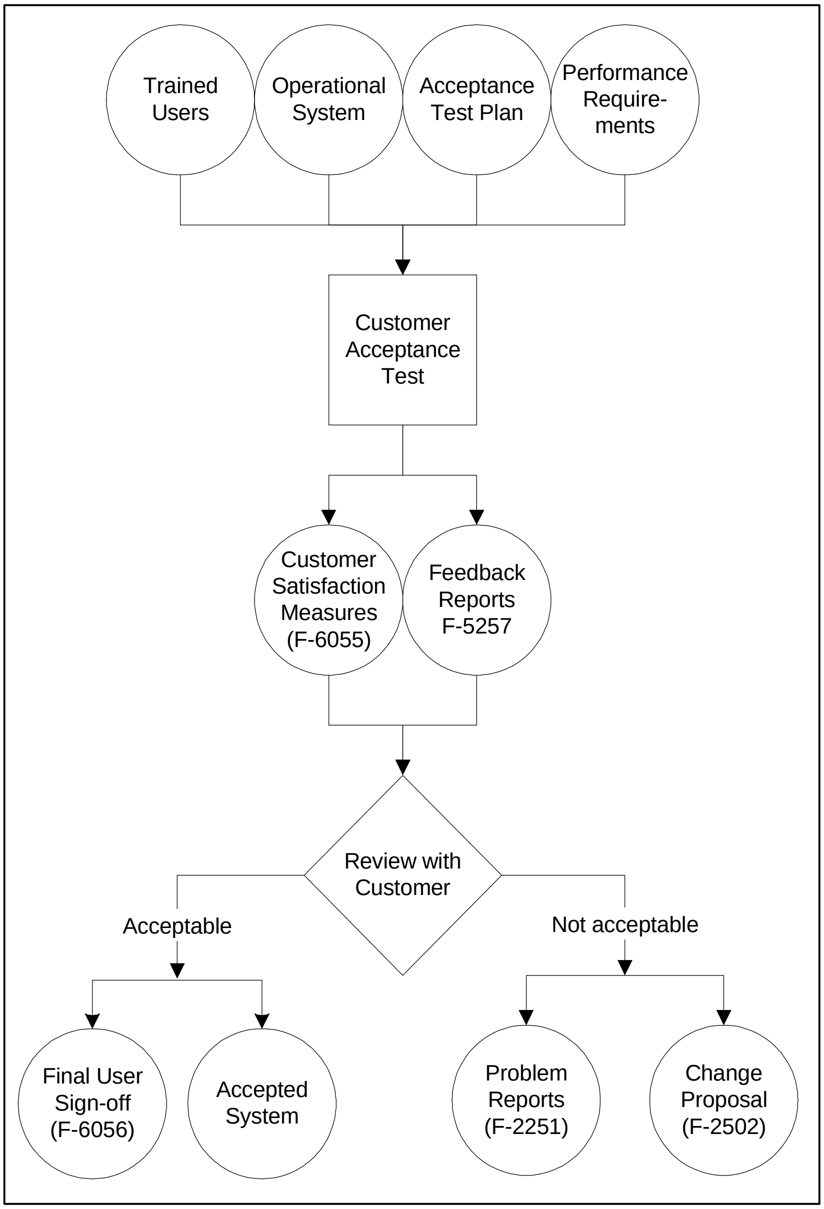


Figure 10–13. Acceptance Test the Solution

Activity 5.13: Analyze the Acceptance Test Results

See Figure 10–14. Using the results of the acceptance test of the system:

1. Review the test results. Log your analysis in a feedback report.
2. If there are discrepancies, log them as problem reports (Forms F–2251 and F–2252) to be resolved in accordance with proper procedures.
3. All appropriate entities evaluate project readiness for production. Appropriate entities must include:
   - Overall Project Manager
   - Technical Project Manager
   - Operational Support
4. If there are no discrepancies, then the system is ready to be moved to production. Secure approvals on Form F-4051 prior to moving the system to the next activity.

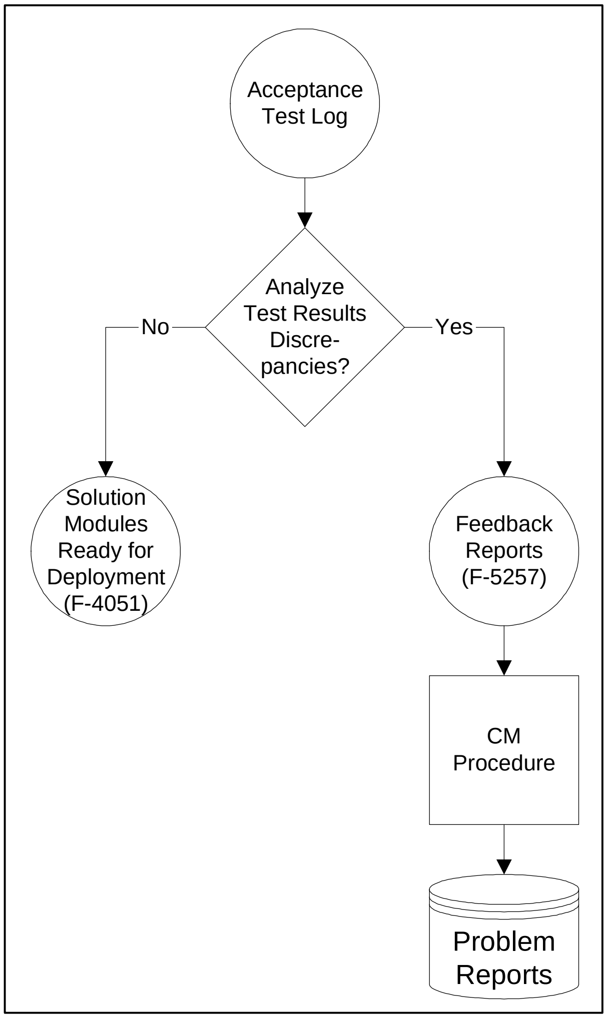

Acceptance Test Log
Analyze Test Results Discre-pancies?
No
Yes
Solution Modules Ready for Deployment (F-4051)
Feedback Reports (F-5257)
CM Procedure
Problem Reports

Figure 10–14. Analyze the Acceptance Test Results

Activity 5.14: Validate and Upgrade the Production Environment

See Figure 10–15. Using the TIP:

1. Validate the production environment (workstation and host or client and server) to determine what the system needs to run and whether the system will run in each individual user environment.
2. If needed, procure and install any additional hardware and software.
3. Upgrade the production environment.

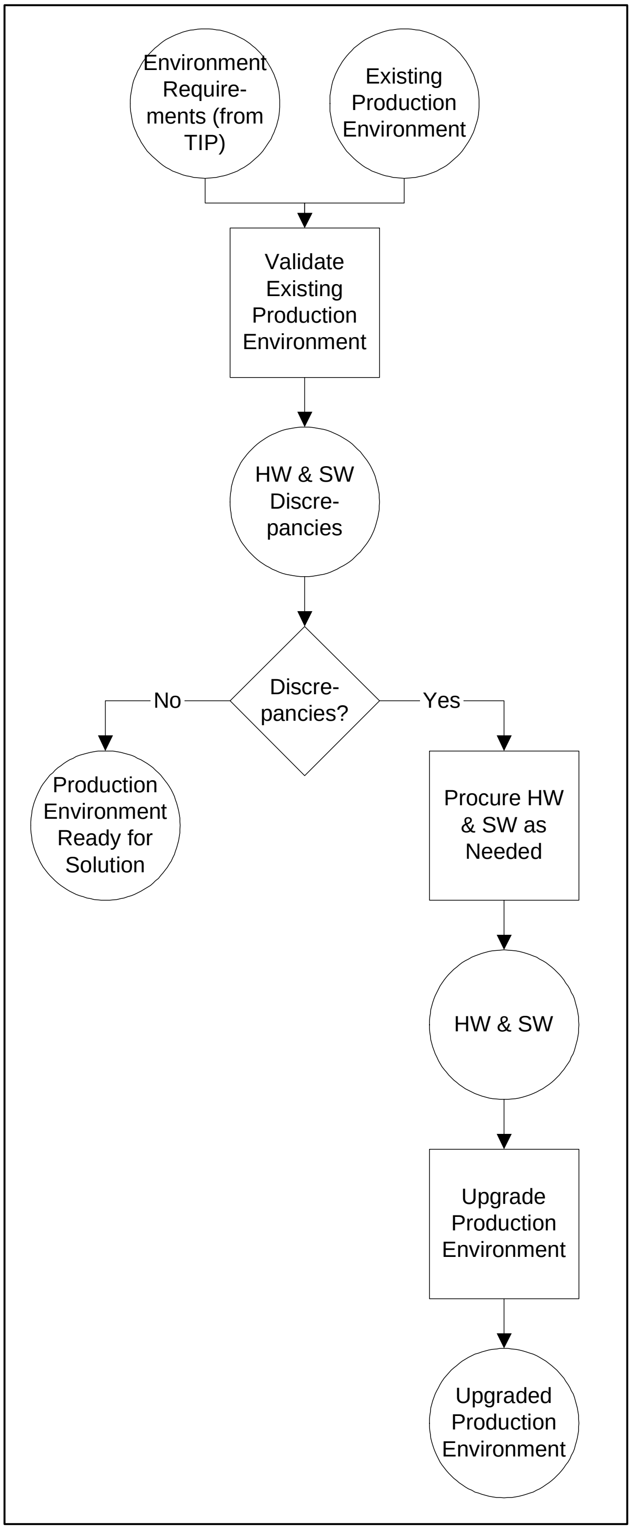

Environment Require-ments (from TIP)
Existing Production Environment
Validate Existing Production Environment
HW & SW Discre-pancies
Discre-pancies?
No
Yes
Production Environment Ready for Solution
Procure HW & SW as Needed
HW & SW
Upgrade Production Environment
Upgraded Production Environment

Figure 10–15. Validate and Upgrade the Production Environment

Activity 5.15: Deploy the Solution to Production Environment

See Figure 10–16. After the system has been acceptance tested, the acceptance test approved and the production environment validated and upgraded:

1. Provide it for operational use in the production environment.
2. Prepare a Rollout Report (F–5255).

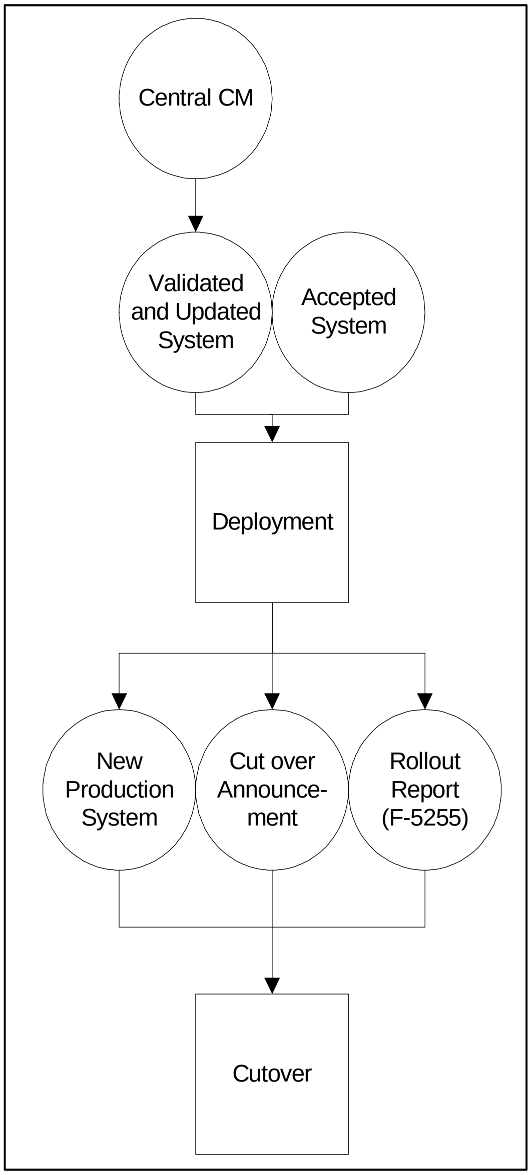


Figure 10–16. Deploy the Solution to Production Environment

Activity 5.16: Review (Component 5)

See Figure 10–17. After the system has been installed, tested, and approved for production use, review the status to decide whether to move into the Component 6, Service the Solution. The outcomes of this evaluation are that the system will either move to Component 6 or, should there be problems precluding that movement, the system moves back to Component 3 for rework to correct the problems.

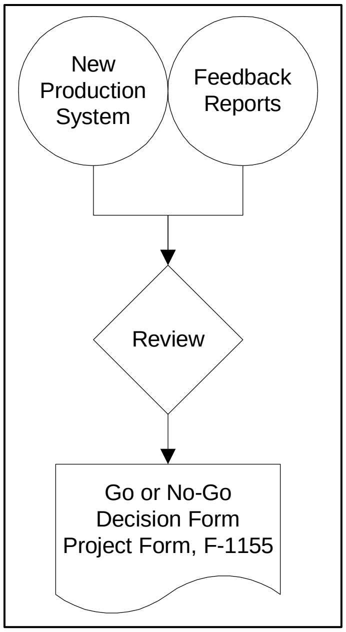


Figure 10–17. Review (Component 5)

11. Component 6. Service the Solution

11.1 Component Overview

The purpose of Component 6, Service the Solution, is to maintain or enhance an existing application or its support environment.

This component is necessarily *very* different from the first five components. The presentation in this chapter is also necessarily very different.

Figure 11–1 provides a high-level illustration of the structure of Component 6. Three different approaches to servicing a solution are shown within the broken line at the bottom of the figure:

- Enhancement
- Release-based maintenance
- Emergency maintenance

Those three approaches are supported by two illustrated elements of configuration management:

- Change management
- Release management

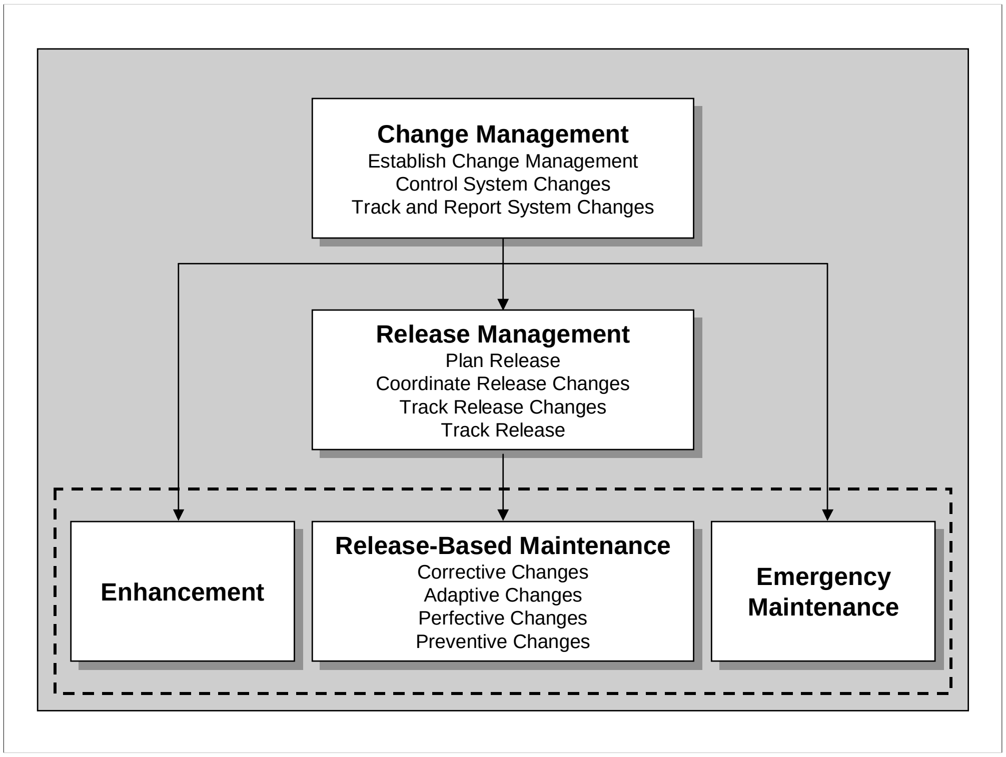


Figure 11–1. High-Level View of Servicing the Solution

Change management evaluates Change Proposals (which may be contractual requests for application system modifications) and Problem Reports and routes them appropriately.

Emergency changes are routed to emergency maintenance. Routine changes are allocated to releases and proceed to release planning. A major change is treated as an enhancement.

Release management takes a defined release, plans the release, and manages changes to the release during its development. One result of planning the release is its division into work packages for assignment to different teams.

Enhancement typically requires a major change to some part of an existing application system, perhaps to the system architecture or to the support environment. Often it is a customer who declares a change to be an enhancement. *Within the SDLCM Methodology, an enhancement is treated the same as a new development effort; that is, by beginning with Component 1 and proceeding through the first five components.* This approach provides adequate visibility to the enhancement Project and ensures that the system documentation will be updated to reflect the major change. The enhancement process is, therefore, not addressed further in this chapter.

Release-based maintenance is the process of taking a defined work package and defining, executing, and testing the required changes.

Emergency maintenance is an alternative to release-based maintenance for changes requiring immediate action. These types of changes, which are the exception rather than the rule, are followed up with a normal release-based maintenance process.

Figure 11–2 provides a process flow diagram for the top level activities of Component 6 showing the triggers and results. This figure also shows that the output of release-based maintenance flows into integration and then to deployment. Integration is the process of taking the defined work package and integrating it with any other work packages to be deployed. Deployment refers to installing and rolling out the integrated solution. Both are discussed in other chapters of this handbook.

The remainder of this chapter is organized around the major processes of change management, release management, release-based (routine) maintenance, and emergency maintenance. An iconic representation of Figure 11–1 appears at the beginning of each section as a reminder of the relationships among the major processes. For each major process, the following are examined and discussed:

- Activities performed
- Roles that perform those activities
- Products produced
- Critical dependencies
- Configuration management of the products
- Quality assurance

Roles, activities, products, and flow are summarized in:

- Process flow diagrams
- Data flow diagrams
- Activity to work product matrices
- Activity role matrices

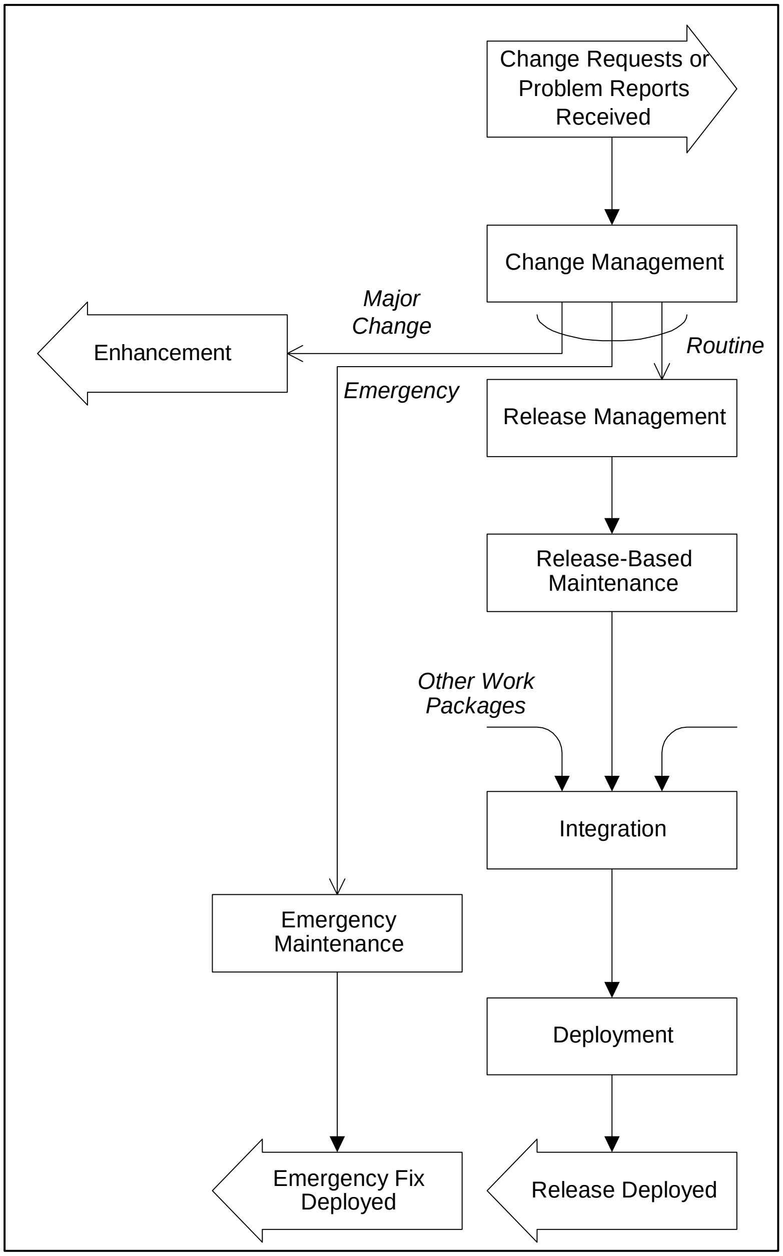
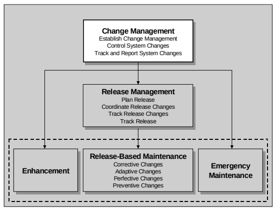


Figure 11–2. Component 6 Process Flow Diagram

11.2 Change Management



This group of activities supports the management of changes to existing systems, including software, hardware, facilities, and staffing. There are three major activities within change management summarized in Figure 11–3:

- Establish Change Management
- Control System Changes

- Track and Report System Changes

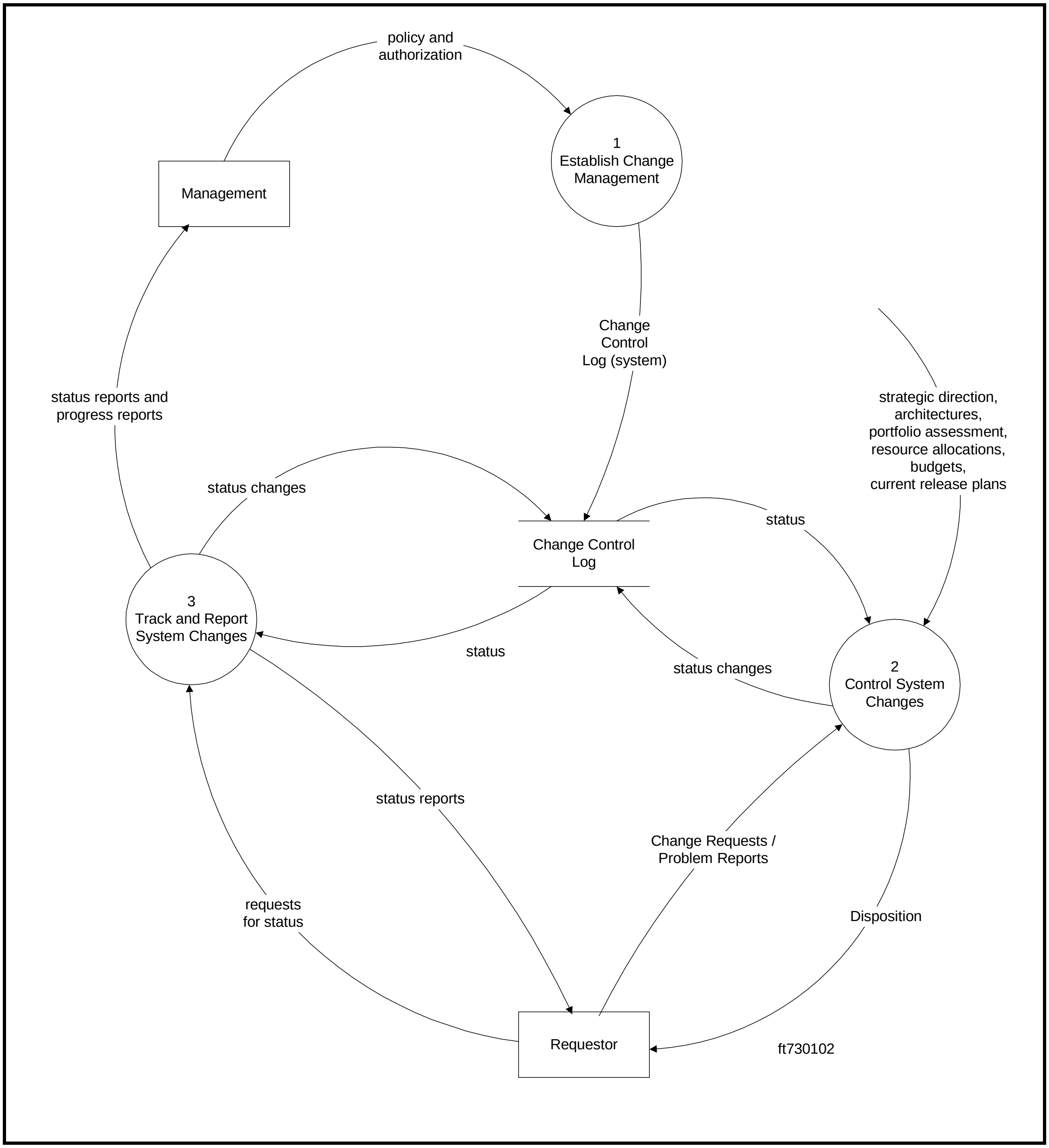

policy and authorization
1
Establish Change Management
Management
Change Control Log (system)
status reports and progress reports
strategic direction, architectures, portfolio assessment, resource allocations, budgets, current release plans
status changes
status
Change Control Log
3
Track and Report System Changes
status
status changes
2
Control System Changes
status reports
Change Requests / Problem Reports
requests for status
Disposition
Requestor
ft730102

Figure 11–3. Change Management Data Flow Diagram

Roles required for change management include:

- **Configuration Control Board (CCB).** This is any group or individual with authority to review and approve changes. There may be a single CCB or multiple CCBs to handle changes of different scope or magnitude. A CCB may be a single individual. (See the Configuration Management chapter for more information.)

- **Configuration Management Office (CMO).** This is the administrative arm of the CCB. This may be one person or a group.
- **Systems engineering.** This group of architects and analysts on the development or maintenance team evaluates the technical requirements of a change proposal or problem report and estimates the effort, cost, and schedule to implement it.
- **Requester.** This is anyone who requests a change or requests information regarding the status of a change. It may be a user, developer, operator, manager, or anyone else with an interest in the system.
- **Management.** This is meant to include anyone with global interest in the change management process and progress in clearing change proposals and problem reports. It may include the CCB, executive management, business function managers, and others.

The following three subsections discuss the steps involved in the Change Management activities and the roles and products associated with those activities.

11.2.1 Establish Change Management

Establishing change management involves establishing the organizations, roles, responsibilities, processes, policies, procedures, tools, and standards to control changes to existing systems. System changes must be controlled through an orderly process to prevent defects and failures and to maximize business value added.

Change management is an integral function of Configuration Management, which is covered in Chapter 5 of this handbook. For the purposes of this chapter, it is sufficient to know that Configuration Management is responsible for performing the following activities, which play a key role in servicing the solution:

- Acquire and set up the tools to support change management, including an automated Change Control Log.
- Develop a Change Proposal form.
- Develop a change management process covering the requesting, evaluating, approving, allocating, tracking, and reporting of system changes.
- Identify the organizations, roles, and responsibilities for performing the change management process.
- Determine the rules for routing different categories of problems, changes, and issues based on category, size (or effort or cost), and risk.
- Develop standards and procedures for completing change proposals.
- Document the policies, procedures, and standards to implement the change management process.
- Communicate the change management policies, procedures, and standards to the organization.

The outputs of the change management function that help support servicing the solution are:

- Change Control Log (tracking and reporting system)
- Change Proposal form (input document)
- Change Management Policies, Procedures, and Standards

11.2.2 Control System Changes

A report of a problem or a request for a system change may trigger the Control System Change activity as shown in the process flow diagram (see Figure 11–4). The data flow diagram (see Figure 11–5) illustrates the flow of data among the ten lower level activities:

1. **Create and Register Change Proposals**. Initiate the change process by creating and registering Change Proposals. Changes may originate from various sources including users, managers, maintainors, and operators. Changes may originate as a trouble call to a help desk or a completed Change Proposal form.

2. **Research, Validate, and Complete Change Proposals**. Make sure the problem or Change Proposal is well defined, properly classified, and valid; complete the information on the request; and assemble all necessary support information. The important point is to make sure the request is adequately documented without requiring excessive documentation that would slow down the process.

3. **Batch Change Proposals**. Review the Change Proposals to determine whether and how they should be batched into Change Packages for further analysis as a unit. Different reported problems might have the same root cause; they may be symptoms of one underlying problem. Several Change Proposals may be satisfied by one general solution. The idea is to see the larger picture.

4. **Analyze Problems**. Analyze a problem to determine its root cause and to propose potential solutions.

5. **Propose Changes**. Propose the specific technical changes required to implement the Change Proposals. By the end of this activity, you should have described the proposed changes in system terms, but you may not have identified all the various components affected by the change.

6. **Analyze Impacts**. Analyze the impact of a proposed change. If not done already, translate the change into system terms and identify all the components affected by the change. At this point, it is not necessary to specify all the changes required but to get a sense for the size of the effort required—counts may be sufficient—so the CCB will have a sufficient basis for evaluating the merits of the change. Another round of impact analysis will occur early in development, once the requirements have been determined.

7. **Estimate Effort, Cost, and Schedule**. Estimate the effort, cost, and schedule for the change, now that the scope is understood, in order to give the users sufficient information to approve or withdraw a proposed change, or a CCB sufficient information to accept or reject a change.

8. **Assess Cost and Benefit**. Combine the cost, benefit, and schedule information into a cost-benefit analysis that will enable the CCB to review the change as a business decision.

9. **Prepare Change Packages for CCB Review**. Assemble Change Package documentation and route the changes to the proper CCBs for review and action. The criteria used to route Change Packages to different groups should include size, scope,

and urgency.

10. **Review and Allocate Change Packages**. CCBs review the Change Packages for merit and priority and then allocate them to particular releases. In doing this, the CCB may remove individual changes from a Change Package and put them in another Change Package, reject them, or defer them indefinitely without assignment to a release. Likewise, the CCB may reject entire Change Packages or defer them indefinitely without assigning them to a release.

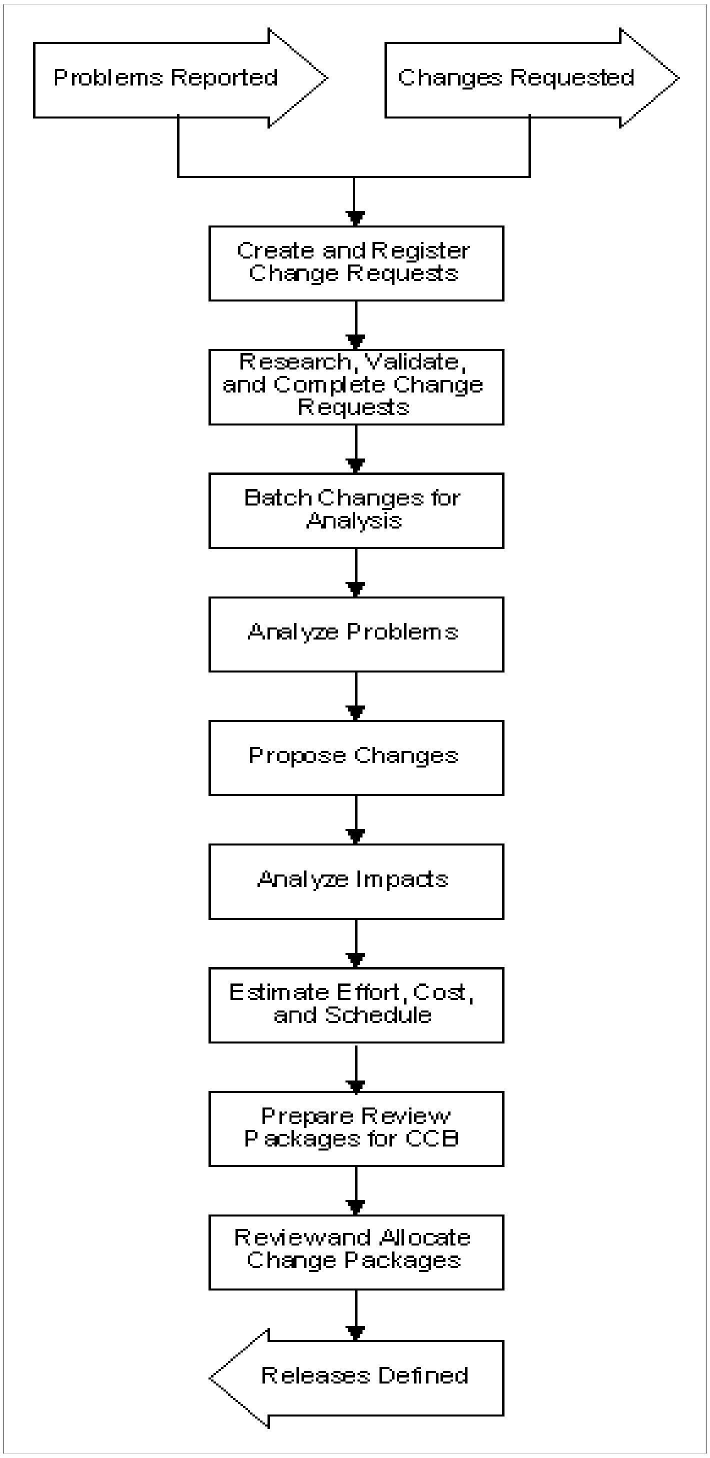

Problems Reported
Changes Requested
Create and Register Change Requests
Research, Validate, and Complete Change Requests
Batch Changes for Analysis
Analyze Problems
Propose Changes
Analyze Impacts
Estimate Effort, Cost, and Schedule
Prepare Review Packages for CCB
Reviewand Allocate Change Packages
Releases Defined

Figure 11–4. Control System Changes, Process Flow Diagram

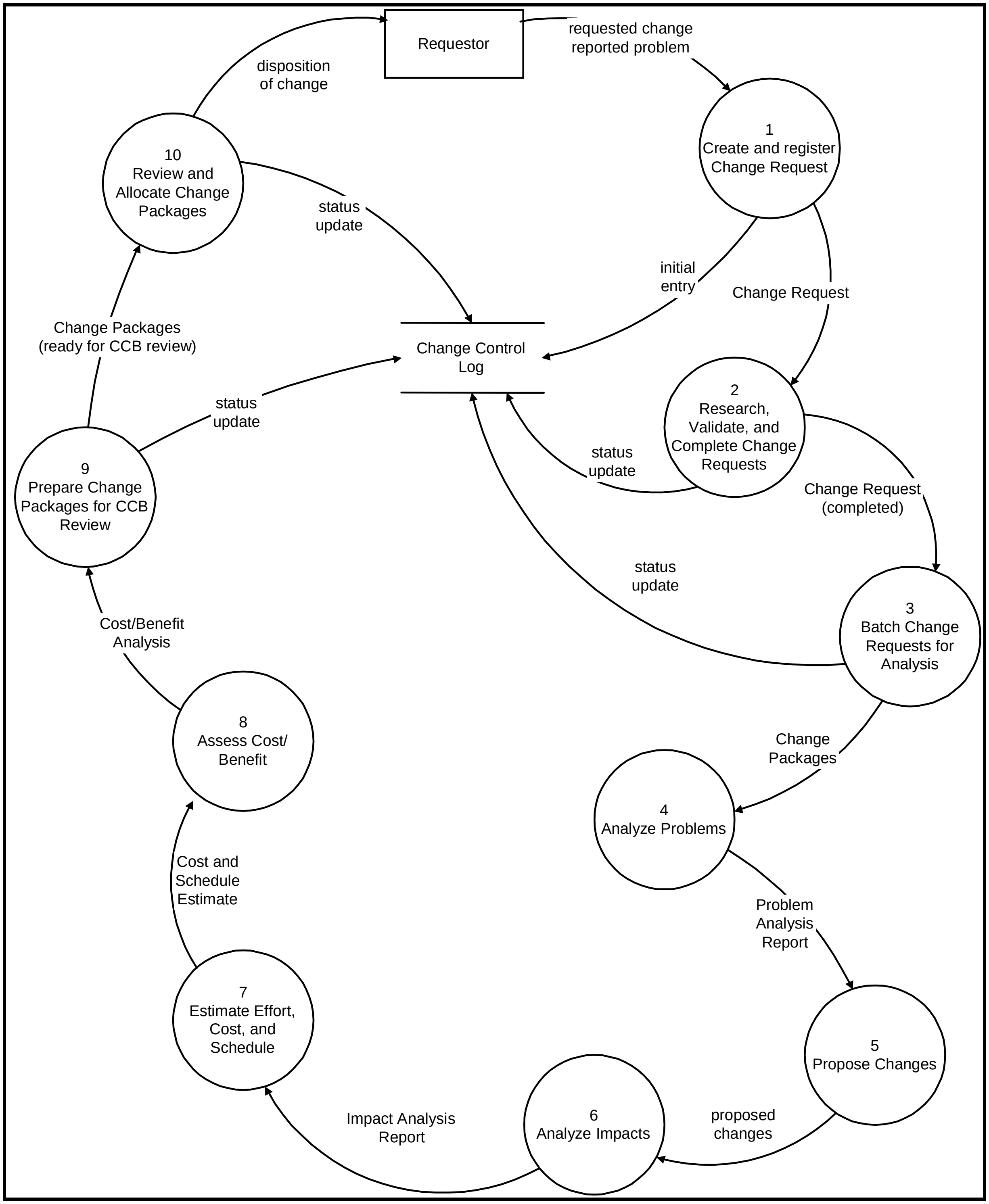


Figure 11–5. Control System Changes, Data Flow Diagram

Table 11–2 associates the ten activities with the roles responsible for their execution.

Table 11–2. Control System Changes, Activity-Role Matrix

Activities	Roles								
Accountable for Execution: Overall Project Manager, Technical Project Manager, Development Team Approval to Proceed: Business Advocate Provides Input: Business SME, Technical SME, Other Representatives or Stakeholders: CM, QA	OPM	TPM	DT	BA	BSME	TSME	OTHER	CM	QA
1. Create and register change proposals.		R	P						
2. Research, validate, and complete change proposals.			P						
3. Batch change proposals.			P						
4. Analyze problems.			P						
5. Propose changes.		R	P						
6. Analyze impacts.			P						
7. Estimate effort, cost, and schedule.		P							
8. Assess cost and benefit.		P							
9. Prepare change packages for CCB review.		P						S	
10. Review and allocate change packages.		P						R	

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

11.2.3 Track and Report System Changes

This activity involves tracking and reporting changes in the status of Change Proposals and Problem Reports and assessing the overall progress in addressing system changes. Triggers and results are shown in the process flow diagram (see Figure 11–6). The data flow diagram (see Figure 11–7) illustrates the flow of data among the three lower level activities:

1. **Track Change Status**. Monitor Change Proposal activity to identify changes in Change Proposal status (either directly as an individual Change Proposal or indirectly through association with a change package or release). Update the Change Log from which the status of individual changes can be obtained and the trends can be analyzed.
2. **Report Change Status**. Report the status of individual changes upon request.
3. **Analyze and Report Change Progress**. Summarize management information from the Change Control Log. Analyze and report trends in the process of addressing Change Proposals. This will show whether the backlog is growing or shrinking and various statistics like mean time to failure, mean time to repair, and so on.

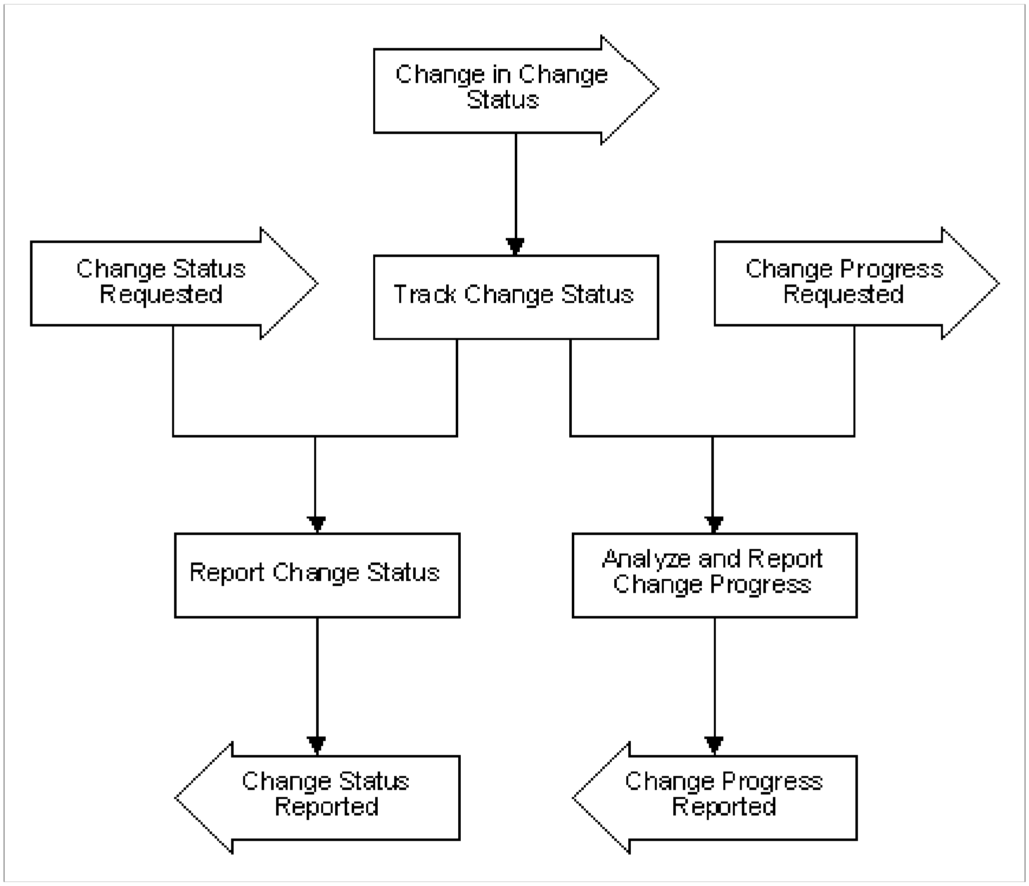


Figure 11–6. Track and Report System Changes, Process Flow Diagram

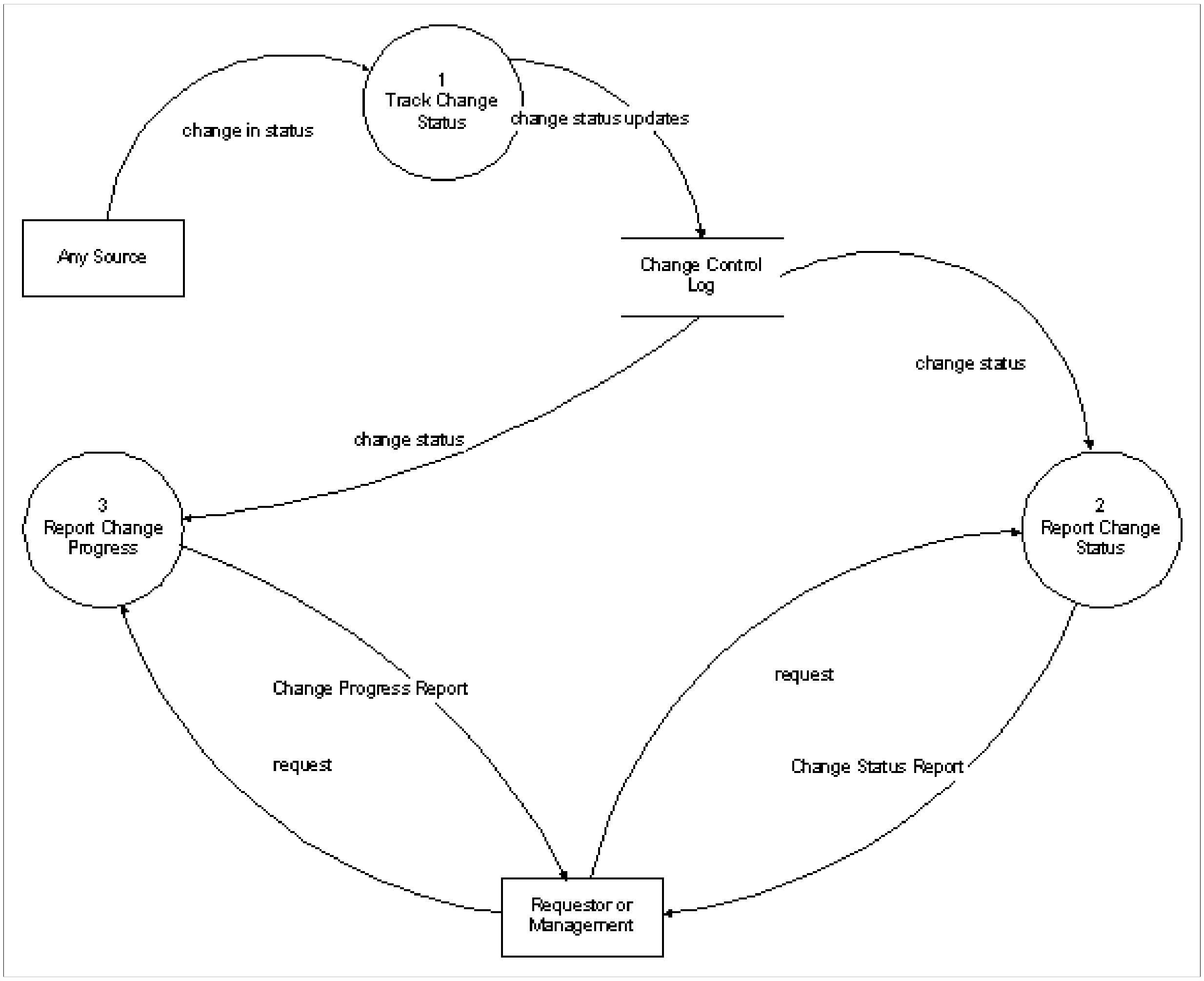


Figure 11–7. Track and Report System Changes, Data Flow Diagram

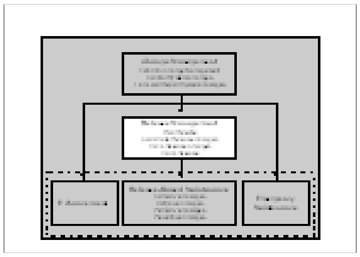
Table 11–3 associates the three activities with the roles responsible for their execution.

Table 11–3. Track and Report System Changes, Activity-Role Matrix

Activities	Roles								
Accountable for Execution: Overall Project Manager, Technical Project Manager, Development Team Approval to Proceed: Business Advocate Provides Input: Business SME, Technical SME, Other Representatives or Stakeholders: CM, QA	OPM	TPM	DT	BA	BSME	TSME	OTHER	CM	QA
1. Track change status.		P							
2. Report change status.		P							
3. Analyze and report change progress.		P						R	

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

11.3 Release Management



The purpose of this group of activities is to manage a release from the time it has been defined through its deployment. It does not include the initial allocation of change packages to a release. Release Management includes four major activities:

- Plan Release
- Coordinate Release Changes
- Track Release Changes
- Track Releases

The following four subsections discuss the steps involved in the Release Management activities and the roles and products associated with those activities.

11.3.1 Plan Release

The purpose of this activity is to plan the release from an engineering point of view, as opposed to a Project management point of view. It verifies the means for development coordination and establishes the structures for managing the components being created or modified by the release.

A defined release triggers this activity as shown in the process flow diagram (see Figure 11–8). There are seven lower level activities:

1. **Verify Development Coordination**. Verify that a development coordination capability exists to coordinate all teams involved in the release and to coordinate the release activities with the larger enterprise. This activity ensures that the organizations, responsibilities, capabilities, working relationships, policies, procedures, and infrastructure exists to provide adequate coordination.
2. **Establish Configuration Management**. Verify and if necessary establish configuration management for the release. Configuration management procedures, tools, and standards should already exist for an application system that is under maintenance, but they should be confirmed. Other configuration management planning is required for each new release.
3. **Review or Establish Standards**. Review all applicable standards to be sure they are adequate for the Project and to create additional standards if needed. Include standards for both products and management documents. Confirm how the standards will be enforced.
4. **Review or Establish Common Services**. The purpose of this activity is to plan for reuse or sharing of components. Review existing reuse policies or establish policies on reuse. Establish what common services are included within the scope of the release, what new common services are required, and what new common services might be contributed from existing components within the scope of the release.
5. **Plan Release Development and Define Work Packages**. Plan how the release should be developed. Divide the release into loosely coupled work packages that can be worked on by separate teams. Scope the work packages to minimize conflicts and

coordination requirements among the teams. Changes that were previously allocated to releases are now allocated to work packages within the release. Any given configuration item is generally assigned to only one work package.

6. **Plan Release Integration**. The purpose of this activity is to plan the integration, integration testing, acceptance, and data conversion testing (if applicable) for the release. This includes any planning for the use of pilots.
7. **Plan Release Deployment**. Plan the deployment of the accepted release to all of the various deployment sites and plan for post-deployment support and operational testing.

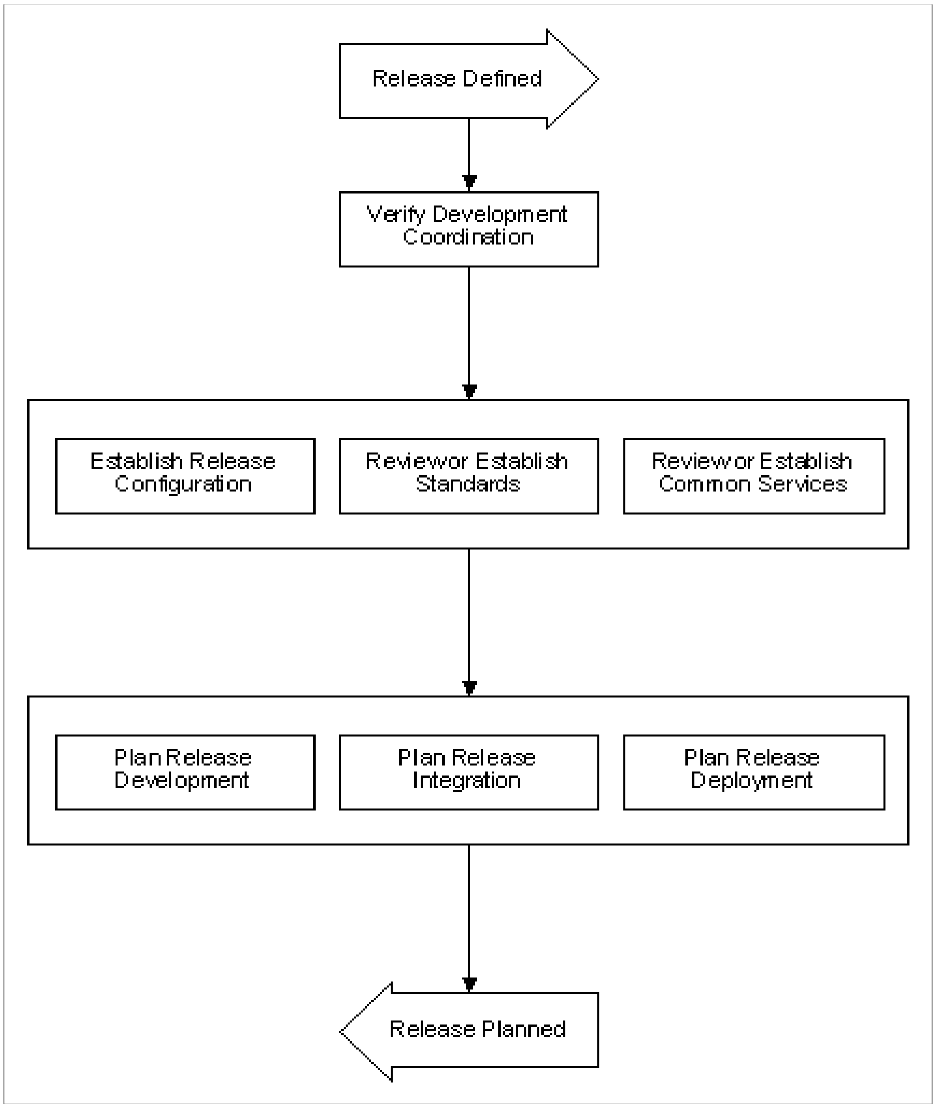

Release Defined
Verify Development Coordination
Establish Release Configuration
Reviewor Establish Standards
Reviewor Establish Common Services
Plan Release Development
Plan Release Integration
Plan Release Deployment
Release Planned

Figure 11–8. Plan Release, Process Flow Diagram

Table 11–4 associates the seven activities with the roles responsible for their execution.

Table 11–4. Plan Release, Activity-Role Matrix

Activities	Roles								
Accountable for Execution: Overall Project Manager, Technical Project Manager, Development Team Approval to Proceed: Business Advocate Provides Input: Business SME, Technical SME, Other Representatives or Stakeholders: CM, QA	OPM	TPM	DT	BA	BSME	TSME	OTHER	CM	QA
1. Verify development coordination.	P	S							
2. Establish configuration management.								P	
3. Review or establish standards.		P							
4. Review or establish common services.		P							
5. Plan release development and define work packages.		P							
6. Plan release integration.		P							
7. Plan release deployment.	P	S							

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

11.3.2 Coordinate Release Changes

The purpose of this activity is to maintain the release plans and to coordinate changes within the release among teams and with enterprise models, standards, plans, and such. Changes are of two types: release plan changes and engineering changes. Release plan changes are changes to release scope, schedule, or cost. Engineering changes are changes to common models, standards, common services, interfaces, common databases, or common infrastructure. Engineering changes may be proposed for use within the release (such as what version of a particular product will be used) or may be proposed by the release for enterprise-wide use, such as a suggested change to the corporate data model.

This activity is triggered when a change proposal is routed to routine released-based maintenance. (See Figure 11–9.) There are nine lower level activities:

1. **Create and Register Change Proposal**. The purpose of this activity is to formalize a requested change to enable its analysis and resolution and to log the change to permit its tracking and prevent its being forgotten.

2. **Analyze Impacts**. The purpose of this activity is to analyze the impact of a requested change. What would have to change to implement the request? How does the change affect other teams working on the release? How does it affect release cost, schedule, or product quality? Does it require a change at the enterprise level?

3. **Negotiate Technical Changes**. The purpose of this activity is to negotiate changes where there is a difference of opinion, for example, if a Project wants a data model change that corporate data administration is fighting.

4. **Estimate Effort, Cost, and Schedule**. The purpose of this activity is to estimate the effort, cost, and schedule required for implementing the change.

5. **Reassess Cost and Benefit**. The purpose of this activity is to assess the impact that the change would have on the cost and benefit measures for the release. This is

intended to provide enough information for the CCB to make a business decision on the proposed change. This activity is performed only if there is an impact on release cost, schedule, or content. Many engineering changes will not affect cost, schedule, or content and thus will not require CCB review.

6. **Package Requests for CCB Review**. The purpose of this activity is to prepare the Change Proposals for presentation to the CCB. Present the facts and required decisions clearly and concisely. Present the questions in order of their importance.

7. **Negotiate and Approve Release Changes**. The purpose of this activity is to negotiate changes to the release plans. Such changes might include a slippage of the date to preserve content or a removal of content to preserve the date.

8. **Implement Changes to Plans**. The purpose of this activity is to implement the agreed-upon changes to the various plans, including plans at the enterprise, release, and Project levels. The result should be a consistent set of plans.

9. **Implement Engineering Changes**. The purpose of this activity is to implement the agreed-upon changes to the various common engineering documents and components. These include changes to models, standards, common services, interfaces, databases, and infrastructure at the individual, team, Project, release, and enterprise levels. The results should be a consistent set of engineering documents and shared components.

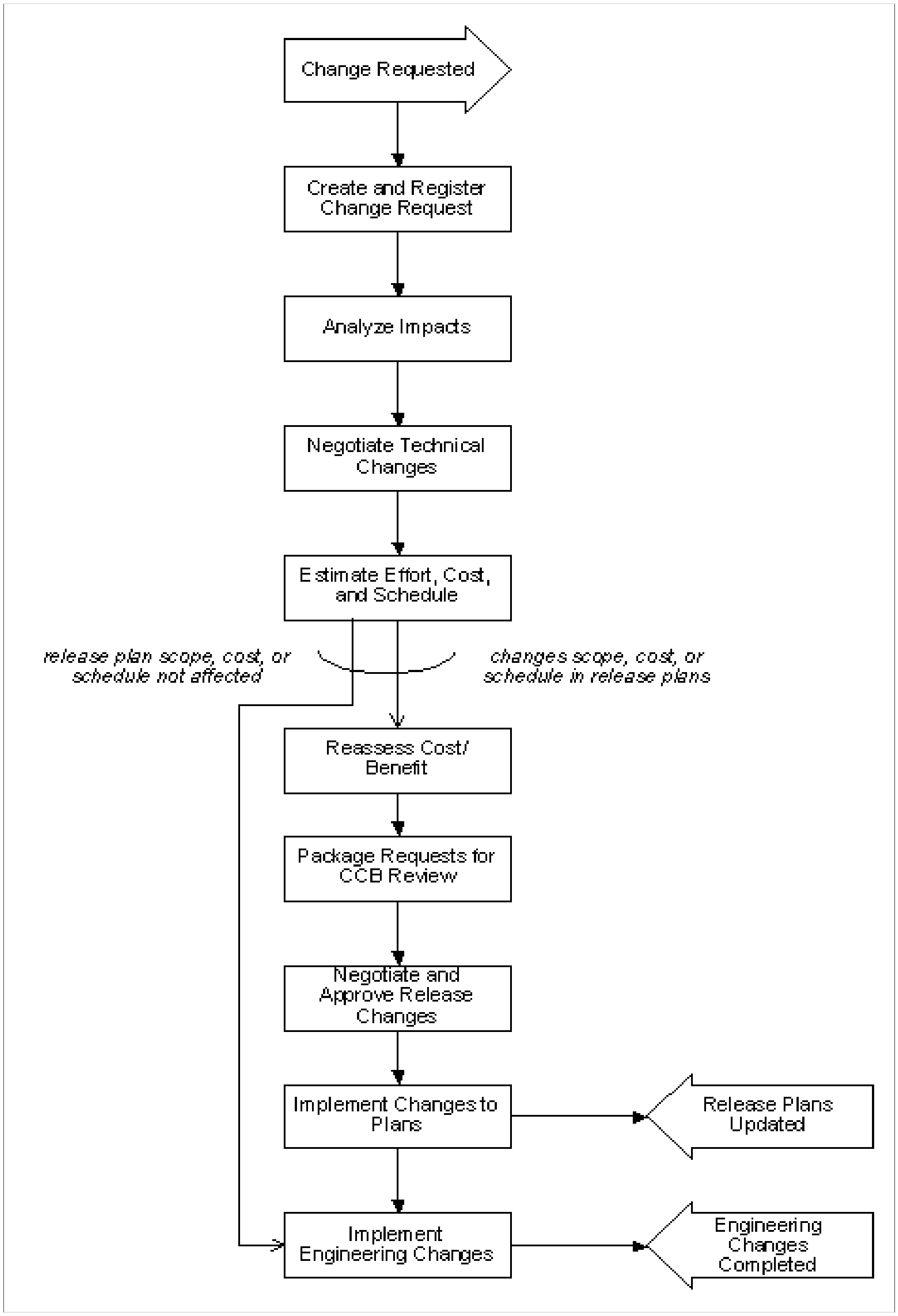

Change Requested
Create and Register Change Request
Analyze Impacts
Negotiate Technical Changes
Estimate Effort, Cost, and Schedule
release plan scope, cost, or schedule not affected
changes scope, cost, or schedule in release plans
Reassess Cost/ Benefit
Package Requests for CCB Review
Negotiate and Approve Release Changes
Implement Changes to Plans
Release Plans Updated
Implement Engineering Changes
Engineering Changes Completed

Figure 11–9. Coordinate Release Changes, Process Flow Diagram

Table 11–5 associates the nine activities with the roles responsible for their execution.

Table 11–5. Coordinate Release Changes, Activity-Role Matrix

Activities	Roles								
Accountable for Execution: Overall Project Manager, Technical Project Manager, Development Team Approval to Proceed: Business Advocate Provides Input: Business SME, Technical SME, Other Representatives or Stakeholders: CM, QA	OPM	TPM	DT	BA	BSME	TSME	OTHER	CM	QA
1. Create and register change proposal.		P							
2. Analyze impacts.		R	P						
3. Negotiate technical changes.	P	P	P						
4. Estimate effort, cost, and schedule.		P	S						
5. Reassess cost and benefit.	P	P							
6. Package requests for CCB review.		P	S					R	
7. Negotiate and approve release changes.		P						P	
8. Implement changes to plans.		P							
9. Implement engineering changes.			P						

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

11.3.3 Track Release Changes

The purpose of this activity is to track and report changes in the status or change proposals and to assess the overall progress in addressing change proposals.

This activity is triggered when a change proposal changes status or there is a request to report status or progress of changes. (See Figure 11–10.) There are three lower level activities:

1. **Track Change Status**. The purpose of this activity is to monitor change proposal activity to identify changes in change proposal status (either directly as an individual Change Proposal or indirectly through association with a change package or release). This updates the Change Log from which the status of individual changes can be obtained and trends analyzed. The Change Report and Change Log for these changes can be the same as used for changes requested for existing systems.

2. **Report Change Status**. The purpose of this activity is to report the status of individual changes upon request.

3. **Analyze and Report Change Progress**. The purpose of this activity is to analyze and report trends in the process of addressing change proposals. This will show whether the backlog is growing or shrinking, among other things.

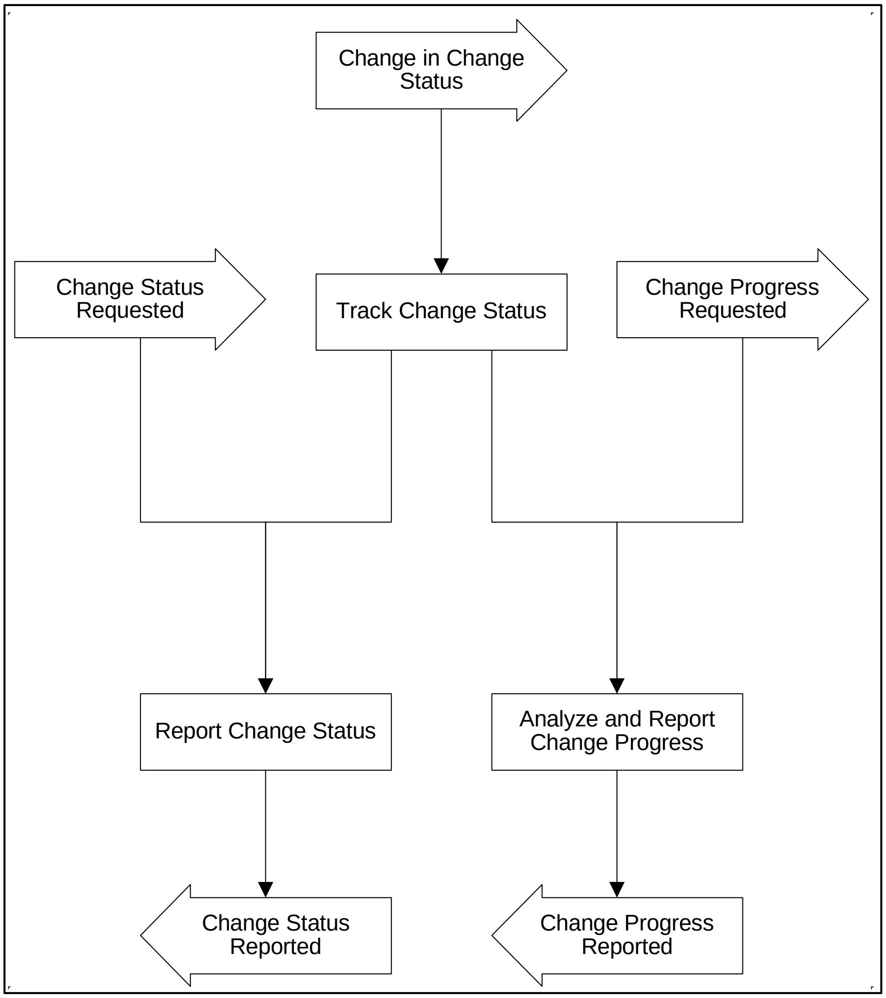


Figure 11–10. Track Release Changes, Process Flow Diagram

Table 11–6 associates the three activities with the roles responsible for their execution.

Table 11–6. Track Release Changes, Activity-Role Matrix

Activities	Roles								
<u>Accountable for Execution</u>: Overall Project Manager, Technical Project Manager, Development Team <u>Approval to Proceed</u>: Business Advocate <u>Provides Input</u>: Business SME, Technical SME, Other Representatives or Stakeholders: CM, QA	OPM	TPM	DT	BA	BSME	TSME	OTHER	CM	QA
1. Track change status		P							
2. Report change status		P	S						
3. Analyze and report change progress		P						R	

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

11.3.4 Track Releases

The purpose of this activity is to track the status and progress of releases. It is fed by the Project status reporting from the Project management processes. A Project can span several releases, and a release can span several Projects. The primary information monitored is the estimated completion date and the estimated cost at completion for the release. This is needed to be able to report the expected deployment dates for changes being implemented by the release. (There may also be intermediate milestones defined for the release, but these are more important for Project management and program assurance than for change management.)

This activity is triggered when a release changes status or there is a request to report status or progress of changes. (See Figure 11–11.) There are three lower level activities:

1. **Track Release Status**. The purpose of this activity is to monitor the planned cost, schedule, and content of the release as it changes based on agreed-upon requested changes and actual events. At any point in time, you should be able to identify the Projected completion date, the Projected cost, and the planned content of the release (that is, which change packages are included).

2. **Report Release Status**. The purpose of this activity is to report, upon request or regular schedule, the Projected completion date, the Projected cost, and the planned content of the release.

3. **Analyze and Report Release Progress**. The purpose of this activity is to report, upon request or regular schedule, any information showing rates of progress. This might be stated in terms of earned value if this approach has been implemented.

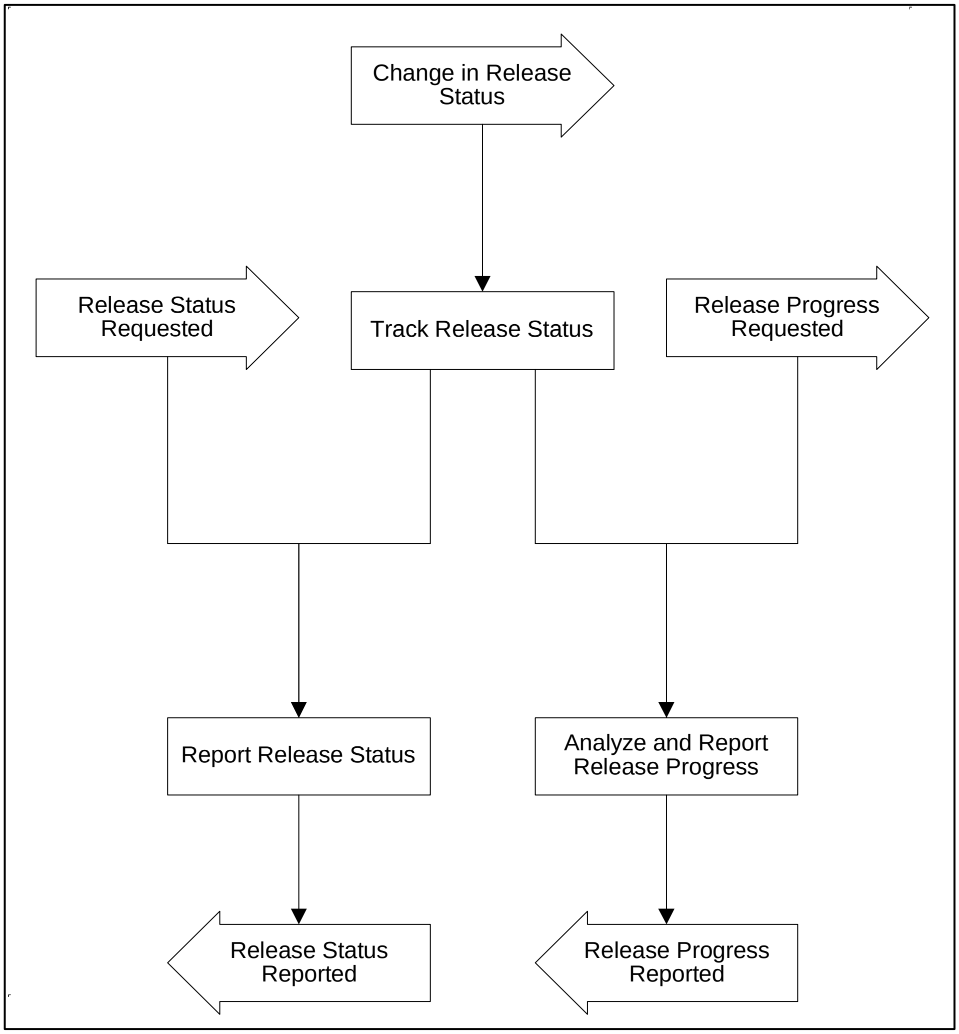


Figure 11–11. Track Releases, Process Flow Diagram

Table 11–7 associates the three activities with the roles responsible for their execution.

Table 11–7. Track Releases, Activity-Role Matrix

Activities	Roles								
Accountable for Execution: Overall Project Manager, Technical Project Manager, Development Team Approval to Proceed: Business Advocate Provides Input: Business SME, Technical SME, Other Representatives or Stakeholders: CM, QA	OPM	TPM	DT	BA	BSME	TSME	OTHER	CM	QA
1. Track release status.		P							
2. Report release status.		P	S						
3. Analyze and report release progress.		P						R	

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

11.4 Release-Based Maintenance

The Routine Maintenance process addresses maintaining and evolving existing systems. It focuses on making changes to an existing environment, taking the operational system and its infrastructure as constraints: the changes have to be built on the underlying structure and have to be compatible with the existing system architecture.

There are four types of changes:

- Corrective changes
- Adaptive changes
- Perfective changes
- Preventive changes

Routine Maintenance may be performed continually and in parallel with other development paths. Triggers and results are shown in the process flow diagram (see Figure 11–12). The data flow diagrams (see Figure 11–13 and Figure 11–14) illustrates the flow of data among the eighteen lower level activities:

1. **Determine Type of Change**. Understand the reason for the change in order to determine the type of change. The type of change information helps you to select the appropriate sequence of activities and their level of detail. Do this for each change contained in the Work Package. This activity starts the Release-Based Maintenance Process and is usually performed after a Work Package has been received. This activity will be completed when each change has been classified as *corrective* or *non-corrective* (that is, adaptive, perfective, preventive). Group changes into sets of related changes.

2. **Extend Requirement Analysis**. Extend the Requirements Analysis to understand the need that caused the Change Proposal. Refine to a more detailed level to produce input for subsequent design activities.

3. **Extend Problem Analysis**. Refine the initial Problem Analysis that was done after receiving the Problem Report until you can identify the underlying cause. Keep in mind that the results of this activity will be the basis for designing the change.

4. **Extend Existing System Analysis**. Create, refine, or update the model views or equivalent system documentation needed for System Evolution and Maintenance. Keep in mind that the results of this activity will be needed for designing the change. This activity may be invoked and performed in parallel to ongoing work when additional, new, or more detailed information is needed. This may happen as a result of a change in system functionality or configuration or during Problem or Requirement Analysis.

5. **Design Change at High-Level**. Work out a high-level design for implementing the change. Focus on the logical level of the programs, the interaction among them or the data entities if the changes are in that area. The results of this activity will be the basis for the implementation of the change. Try to keep the design compatible and aligned with the existing architecture and operation.

6. **Review High-Level Design**. Review the High-level Design for the change. Validate that

the design basically adheres to the current architecture and its use and that necessary changes are compatible to them. Approve or update related estimates for resources, schedule, and costs.

7. **Reassess Impact**. Reassess the impact of the change in the context of the current system and the modifications to be done. If necessary, refine the Impact Analysis in order to get to a valid and agreed assessment of the consequences of the change.

   This activity incrementally creates, updates, or refines information on the change's impact upon the existing system architecture. It may be iterated. As Analysis proceeds or resumes, its scope may evolve from deeper to higher level of detail as well as from rather limited to broader scope.

8. **Decide Whether to Proceed**. Decide whether to promote the change to the implementation activity in the proposed way or to turn the change back for reconsideration. At this time, the impact of the change, the resources needed to implement it, and both the schedule and costs estimates have been validated; and they describe the different aspects of the change correctly.

   In the case the change is turned back, there are two choices:

   (1) Create a different design by repeating the prior steps beginning with "Design Change at High-Level" or

   (2) Raise an issue and send the change to back to the Change Management process.

9. **Implement Change**. The following paragraphs describe the activity "Implement Change" in a generalized way that can be tailored to different development approaches.

   Implementation is a composite activity that can vary greatly in size and complexity given a particular problem set, environment, and development approach. However, regardless of the specific development approach chosen, implementation is made of three kinds of basic activities: design, construction, and validation (or design, code, test). The basic implementation approaches are constructed by wiring together these activities in different sequences and iterations, for different units of scope and time, in different technical environments, and with different human dynamics:

   One option is an iterative learning approach, designing the business process change and then deriving the system change, going through several cycles of design, code, and validate, that allow for learning during the development process and uncovering hidden needs.

   Another option is an accelerated approach, going through several tightly scheduled time boxes consisting of design, code, validate in a lab environment, with functionality made secondary to schedule.

   Another option is an incremental waterfall approach, with larger chunks of design, code, and validate, but with greater emphasis of formal validation of requirements and design.

   Another option is a package-based approach, with requirements transmuting into selection criteria and selection becoming part of the design activity (as in adding a COTS module onto an existing application).

   If prototyping is used, go through several iterations of design, construction, and validation of a prototype as a means to discover hidden requirements and to arrive at a design for the target system.

After the prototyping stage is complete, the coding stage completes the armor-plating process if a limited function prototype is being evolved into the target system; but if only simulation prototyping was used, the coding stages is the construction of the target system in the target environment.

If a package-based approach is selected, understand Design as to include package selection as well.

If a requirements-driven approach is selected, create the Design by use of the Requirement Analysis Technique and iteration to more detailed level.

To assemble an implementation approach, take the basic building blocks, apply tailoring to the specific context, scope, and level of detail, and iterate as needed to cover the whole change. This will provide an implementation path that is tailored to the change and the constraints of the existing environment and satisfies the preferences of the customer.

Start each path variant with an enabling activity that establishes the prerequisites in order to perform the implementation activities.

There are four steps for implementing a change:

- Prepare for Implementation. Establish or validate the prerequisites that are needed for implementing the change. Include both technical and organizational aspects to be able to continue work with designing the next level of detail of the change.
- Design Change detailed. Refine the High-Level Design of the change down to the Unit or Module level. The goal of this activity is to develop a physical design for programs and the referenced data so that coding can start. Several iterations may be necessary until this level of detail will be reached.
- Code the Change. Code the change on the basis of the detailed design that has been created before. In parallel to this work, create Test Cases for testing code sections during implementation and later for Unit Testing.
- Short-Test Code (optional). Perform some short tests on selected code sections to get confirmation that the changed modules will basically work and are mature for entering Unit Testing. Short Testing may help to detect obvious implementation or unit design errors very early. The test takes place in the Personal Library. Proceed with coding on the next change item after successful Short test.

10. **Unit Test Change**. When the Implementation has been completed, test the new or changed functions to make sure that they satisfy the requirements and needs described in the Change Proposal or Problem Report. Within Unit Test, focus on the functionality of the change.
11. **Consolidate Changes of Work Package**. Integrate all modules related to the changes of the Work Package with the unchanged modules of the application to be able to perform the Work Package Integration Test.
12. **Perform Work Package Integration Test**. After having integrated all changes of a Work Package, test the new or modified functions of the entire Change Package. Within this test, focus on the communication and interaction between the changes of the Work Package when using the these functions.
13. **Perform Regression Test on Application**. Perform Regression Testing on Application

level to ensure that functionality that should persist has not been affected by the new or modified functions introduced with the Change Package.

14. **Integrate Release**. Integrate all Work Packages of the Release with the remaining unchanged modules of the Release. This activity links together the non-modified components with the new or changes module for the first time. It prepares Integration and Regression Testing for the entire Release.
15. **Test the Release**. After the release has been integrated, this activity determines if the existing parts of the release harmonize with new functionality introduced by the Work Packages. Within this test, focus on the interaction of the subsystems or major components of the system or release when using the new functions.
16. **Perform Regression Test on Release**. Perform Regression Testing on Release level to ensure that the existing functionality of the Release has not been affected by the new or modified functions introduced by the Change Packages.
17. **Plan Testing (ongoing).** Test Planning is an ongoing activity in which new information is added to existing documents as it arises. It proceeds in parallel with the design and implementation process, creating Test Cases as soon as they can be defined. Follow this approach if it is feasible.
18. **Document Changes (ongoing).** Document the change on its way through the development life cycle. This documentation is needed primarily for understanding the change. It also helps to keep the system information up to date and correct.

Figure 11–13. Release-Based Maintenance, Data Flow Diagram (1 of 2)

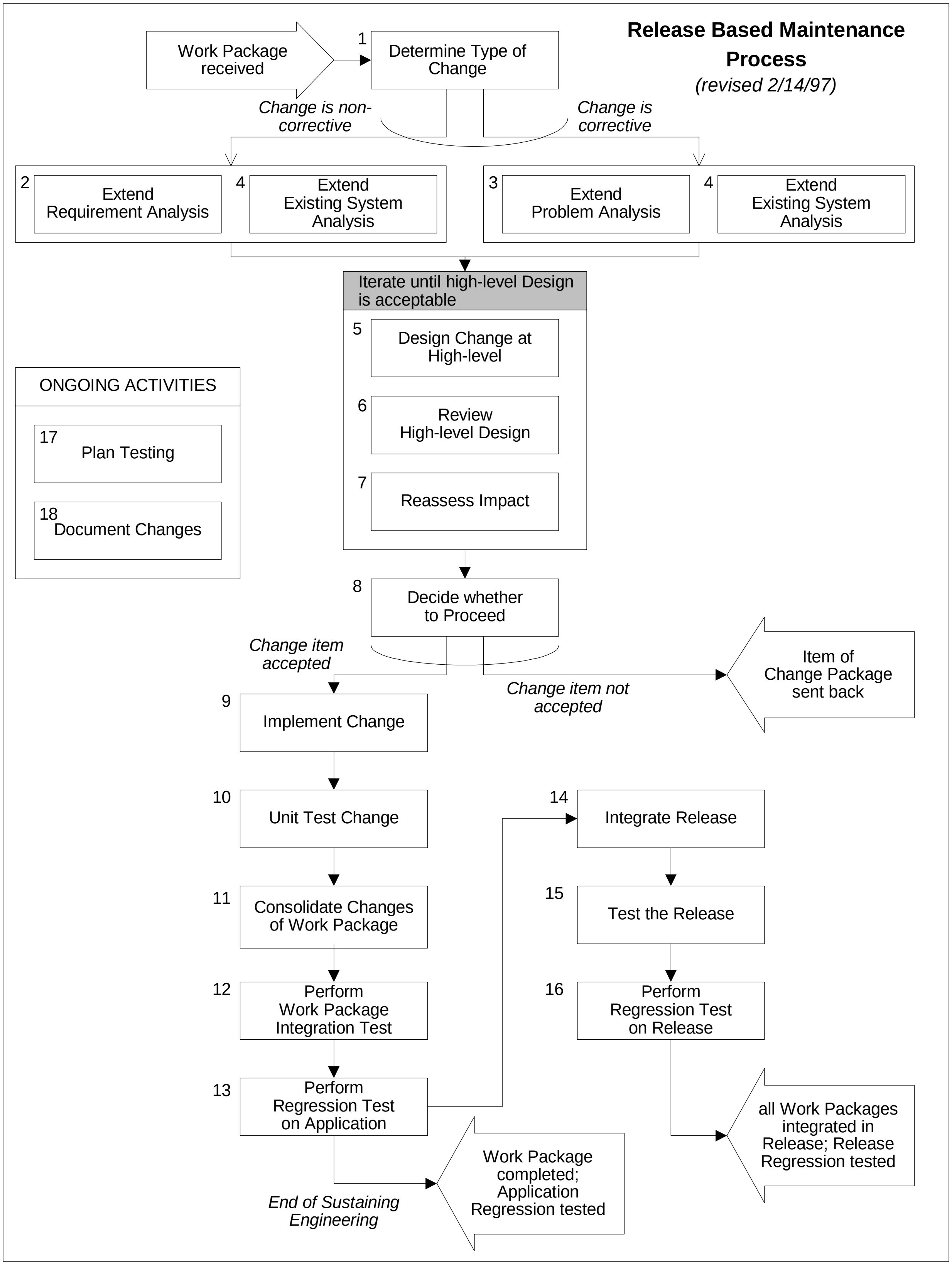

Release Based Maintenance Process
(revised 2/14/97)
Work Package received
1 Determine Type of Change
Change is non-corrective
Change is corrective
2 Extend Requirement Analysis
4 Extend Existing System Analysis
3 Extend Problem Analysis
4 Extend Existing System Analysis
Iterate until high-level Design is acceptable
5 Design Change at High-level
6 Review High-level Design
7 Reassess Impact
ONGOING ACTIVITIES
17 Plan Testing
18 Document Changes
8 Decide whether to Proceed
Change item accepted
Change item not accepted
Item of Change Package sent back
9 Implement Change
10 Unit Test Change
11 Consolidate Changes of Work Package
12 Perform Work Package Integration Test
13 Perform Regression Test on Application
14 Integrate Release
15 Test the Release
16 Perform Regression Test on Release
all Work Packages integrated in Release; Release Regression tested
Work Package completed; Application Regression tested
End of Sustaining Engineering

Figure 11–12. Release-Based Maintenance, Process Flow Diagram

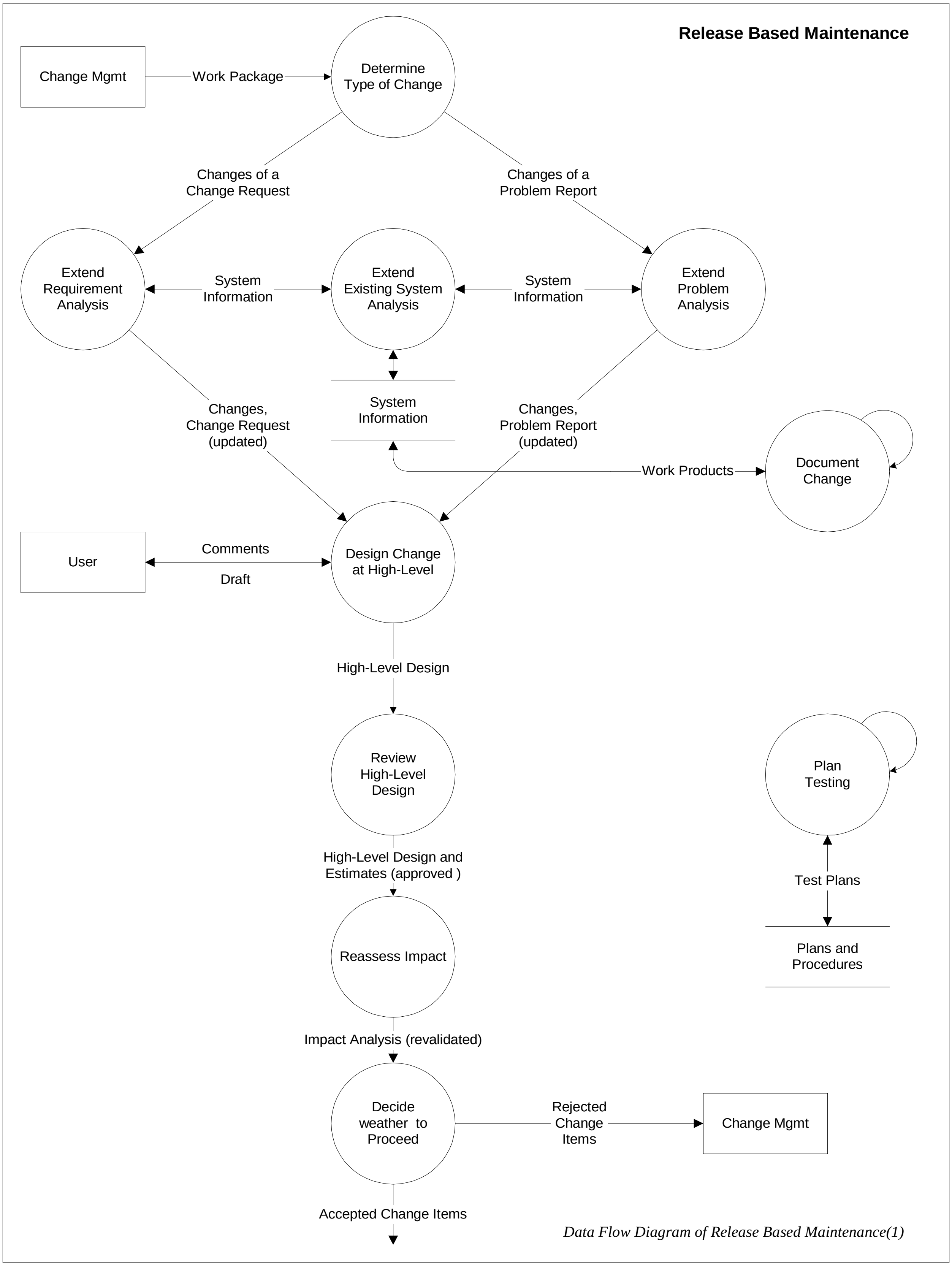

Release Based Maintenance
Change Mgmt
Work Package
Determine Type of Change
Changes of a Change Request
Changes of a Problem Report
Extend Requirement Analysis
System Information
Extend Existing System Analysis
System Information
Extend Problem Analysis
System Information
Changes, Change Request (updated)
Changes, Problem Report (updated)
Work Products
Document Change
User
Comments
Draft
Design Change at High-Level
High-Level Design
Review High-Level Design
Plan Testing
High-Level Design and Estimates (approved)
Test Plans
Plans and Procedures
Reassess Impact
Impact Analysis (revalidated)
Decide weather to Proceed
Rejected Change Items
Change Mgmt
Accepted Change Items
Data Flow Diagram of Release Based Maintenance(1)

Figure 11–13. Release-Based Maintenance, Data Flow Diagram (1 of 2)

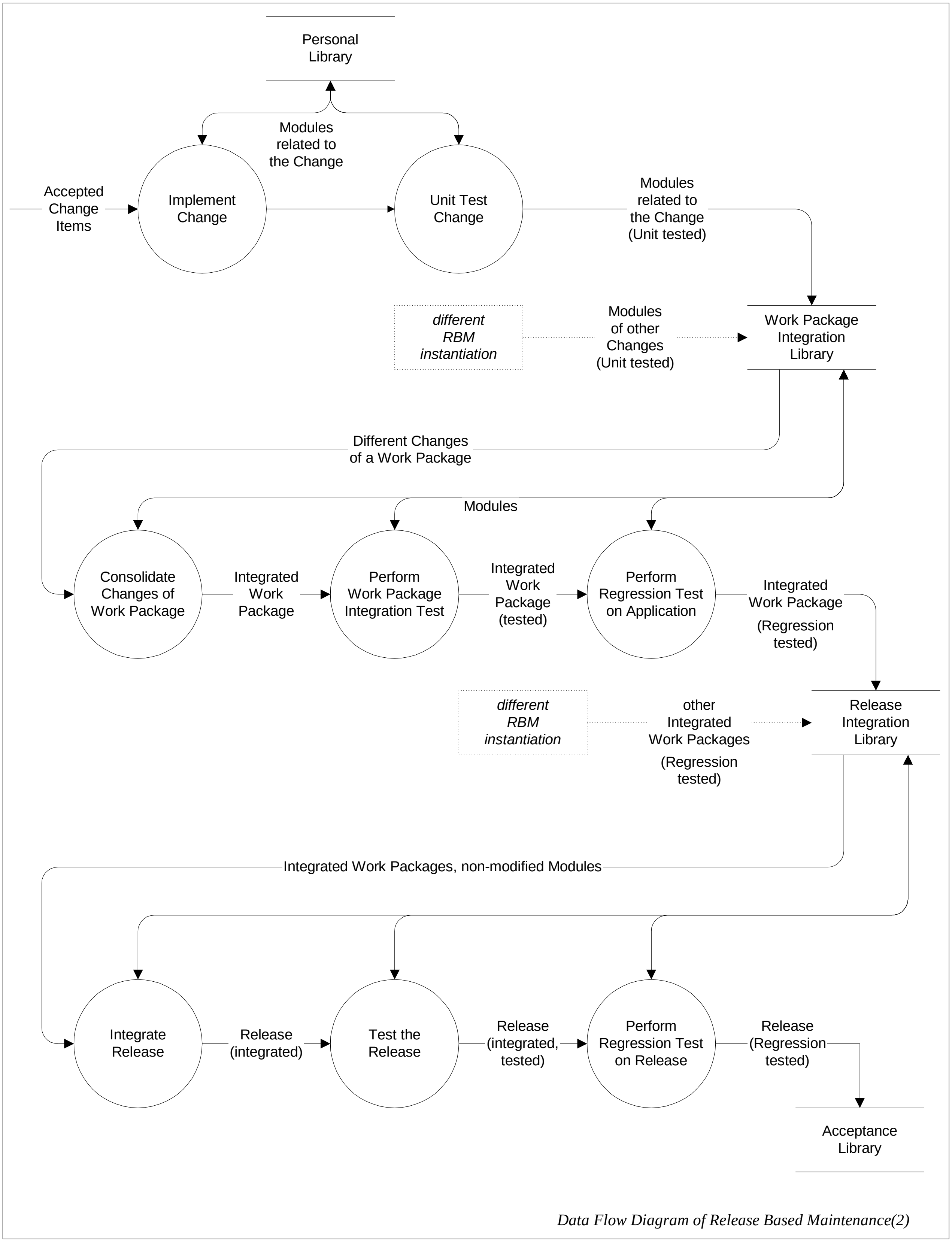

Personal Library
Modules related to the Change
Accepted Change Items
Implement Change
Unit Test Change
Modules related to the Change (Unit tested)
different RBM instantiation
Modules of other Changes (Unit tested)
Work Package Integration Library
Different Changes of a Work Package
Modules
Consolidate Changes of Work Package
Integrated Work Package
Perform Work Package Integration Test
Integrated Work Package (tested)
Perform Regression Test on Application
Integrated Work Package
(Regression tested)
different RBM instantiation
other Integrated Work Packages
(Regression tested)
Release Integration Library
Integrated Work Packages, non-modified Modules
Integrate Release
Release (integrated)
Test the Release
Release (integrated, tested)
Perform Regression Test on Release
Release (Regression tested)
Acceptance Library
Data Flow Diagram of Release Based Maintenance(2)

Figure 11–14. Release-Based Maintenance, Data Flow Diagram (2 of 2)

Table 11–8 associates the 18 activities with the roles responsible for their execution.

Table 11–8. Release-Based Maintenance, Activity-Role Matrix

Activities	Roles								
Accountable for Execution: Overall Project Manager, Technical Project Manager, Development Team Approval to Proceed: Business Advocate Provides Input: Business SME, Technical SME, Other Representatives or Stakeholders: CM, QA	OPM	TPM	DT	BA	BSME	TSME	OTHER	CM	QA
1. Determine type of change.			P						
2. Extend requirement analysis.			P						
3. Extend problem analysis.			P						
4. Extend existing system analysis.			P						
5. Design change at high-level.			P						
6. Review high-level design.		S	P						
7. Reassess impact.		P							
8. Decide whether to proceed.	P								
9. Implement change.			P						
10. Unit test change.			P						
11. Consolidate changes of work package.			P						
12. Perform work package integration test.			P						
13. Perform regression test on application.			P						
14. Integrate release.			P						
15. Test the release.			P						
16. Perform regression test on release.			P						
17. Plan testing (ongoing).			P						
18. Document changes (ongoing).			P						

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

Figure 11–15 depicts the flow of a software module through the various software libraries during the development, integration, testing, and deployment processes. The figure includes the paths for both routine (release-based) maintenance and emergency maintenance (addressed in the next section).

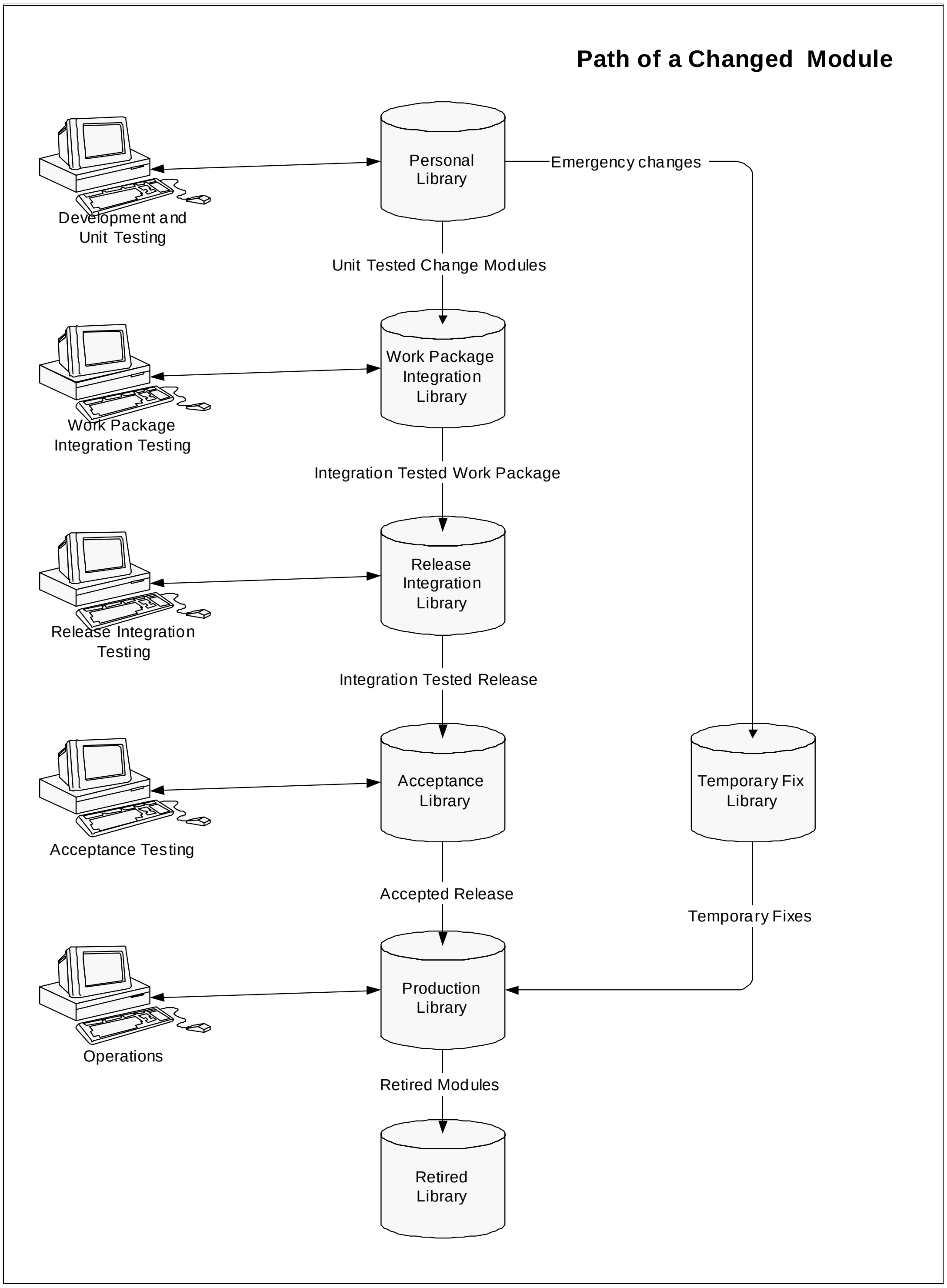

Path of a Changed Module
Personal Library
Emergency changes
Development and Unit Testing
Unit Tested Change Modules
Work Package Integration Library
Work Package Integration Testing
Integration Tested Work Package
Release Integration Library
Release Integration Testing
Integration Tested Release
Acceptance Library
Temporary Fix Library
Acceptance Testing
Accepted Release
Temporary Fixes
Production Library
Operations
Retired Modules
Retired Library

Figure 11–15. Software Libraries for Maintenance

11.5 Emergency Maintenance

Emergency Maintenance addresses maintenance required in emergency situations. An Emergency situation may arise during system operation when system errors with severe impact occur and cannot be handled by the system administration. Such errors must be fixed as quickly as possible to return to operation.

After an emergency situation is fixed by an Emergency Maintenance intervention, the related Problem Report is reconsidered by Change Management and fed into the queue of pending changes to the system. Emergency Maintenance is often a temporary fix to solve an urgent problem very quickly and normally is often not suitable as a lasting change to the system. Thus, it should go into the short list of changes to be included in one of the next releases and then become a permanent and fully quality-assured change to the system.

To ensure consistent decisions with minimal delay, control of the Emergency Maintenance activities is concentrated in an Emergency Management role within Change Management. Emergency Management's primary responsibility is to control the Emergency Maintenance processes and to make the decisions effectively and very fast in an area where system and business issues overlap. It also serves as the interface between Operations and the Emergency Maintenance Team. Emergency Management may consist of only one experienced individual with sound knowledge in both the systems and business worlds. Alternatively, there might be two individuals acting as an efficient team.

It is vital for performing Operations and Emergency Maintenance effectively that information on the system is available and reliable. If such information is not yet available, apply analyze the existing system to establish at least the minimum set of required documentation or refine the existing information if there are gaps before the first emergency situation occurs.

Figure 11–16 gives a high level overview of how Emergency Maintenance fits into the overall cycle of system maintenance and operation. The boxes represent process groups, the arrows the flow of the indicated work products. The figure illustrates the following:

- During Operation, an emergency situation occurs. After a first diagnosis, a Problem Report is sent to Emergency Management.
- Emergency Management assesses the gravity of the problem, different solution approaches, and decides whether an Emergency Maintenance Intervention is needed.
- If it is an instance for Emergency Maintenance, Emergency Management sends the Problem Report to the Emergency Maintenance Team.
- The Emergency Maintenance Team then works to solve the problem by work-arounds or adequate system changes (varies from patches to source code changes).
- After a solution has been identified, implemented, and sufficiently tested, it is fed into production environment.
- Emergency Management (within Change Management) receives the updated Problem Report and documentation on the emergency fix and then decides

whether the change will be promoted into the queue of candidate changes for future releases.

- Emergency Management is responsible that Emergency Maintenance interventions are tracked and reported to Operations and Change Management.

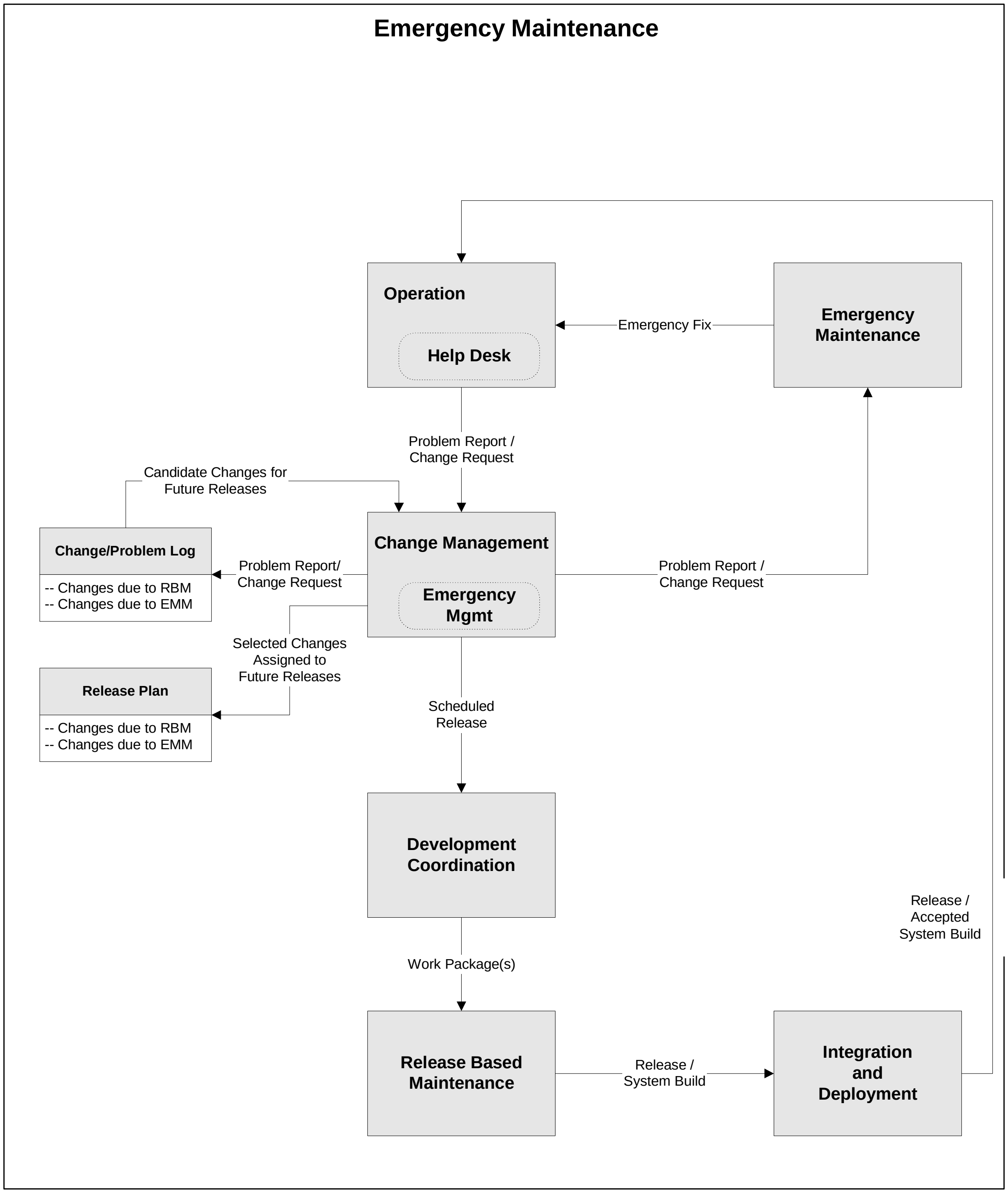


Figure 11–16. Emergency Maintenance in Context

Triggers and results are shown in the process flow diagram (see Figure 11–17). The data flow diagram (see Figure 11–18) illustrates the flow of data among the nine lower level activities:

1. **Understand the Problem**. Study and understand the problem as it has been described by the Operations staff in the Problem Report. Check and complete the information as needed to be able to start a solution approach.

2. **Assess and Classify Problem**. Once the problem is understood, assess the gravity of the problem, the risks, and the impact of performing or omitting an intervention to the system. Classify the problem and decide if an Emergency Maintenance Intervention is required. Determine if the problem can be resolved at the system administration level; if not, proceed with Emergency Maintenance.

3. **Identify Solution Approaches**. With the problem now classified as an Emergency Maintenance case, continue to study the problem to identify alternative solution approaches. Describe the approaches in an Intervention Plan and propose a sequence for choosing the next approach if the prior one has failed.

   Intervention Plan. An Intervention Plan documents the emergency fix approaches together with their priority and possible interdependencies. It describes the sequence of different approaches that have been created to solve the emergency problem.

   The Intervention Plan also serves as a log that captures the complete history of the emergency intervention and the information that is derived from the approaches that failed. This log provides essential lessons learned for subsequent interventions.

   Keep the Intervention Plan short and easy to understand. Remember that it supports the emergency intervention and captures the information about the process.

   Attach the Intervention Plan to the Problem Report so that all records of the emergency maintenance are kept together.

4. **Validate Emergency Maintenance Environment**. After identifying the technical, personnel, and organizational resources required for performing the Emergency Maintenance, ensure that all prerequisites are satisfied at all Operations and Emergency Maintenance Team sites.

   At the completion of this activity, the teams and environments at Operation and Emergency Maintenance Team sites are ready for implementing the solution approaches and testing the emergency fix.

5. **Implement and Test Solution**. Start implementing the solution as soon as a solution approach has been selected. To avoid delays, perform short tests of the modified or new code related to the Emergency Fix as early as possible.

6. **Install Emergency Fix**. Make sure that the Operations site is ready for installing the Emergency Fix and that security provisions can be performed. Install the Emergency Maintenance Fix as soon as it has been implemented, all materials are available, and short-tested.

7. **Test Emergency Fix**. Test the Emergency Fix against the problem description under the same conditions as the problem occurred. Execute the fix at the Operations site. Record the findings in the Intervention Plan.

8. **Evaluate Results of Approach**. After the Fix has been tested, evaluate the results of the solution approach for the emergency problem to decide if the Emergency Fix solves the problem sufficiently or if another approach must be taken.

9. **Reassess Emergency Problem and Fix**. After the emergency situation has been resolved, reassess both the problem that caused that emergency and the Emergency Fix. Include both system and business aspects in the assessment. Transfer the Emergency Fix into a Change Proposal and pass it to the Change Management if this is appropriate. The Fix will then be replaced by a release-based change in a later release.

   Emergency fixes are done under severe time restrictions, with the focus on getting the system to operation as quick as possible and not on the quality aspects that apply to regular system evolution as done by Release-Based Maintenance. Furthermore, emergency fixes might be done as patches to run-time modules and not traced back to the source code modules after the system is running again. This situation may cause increased maintenance efforts, and increase the risk of losing the fix in a subsequent system or releases.

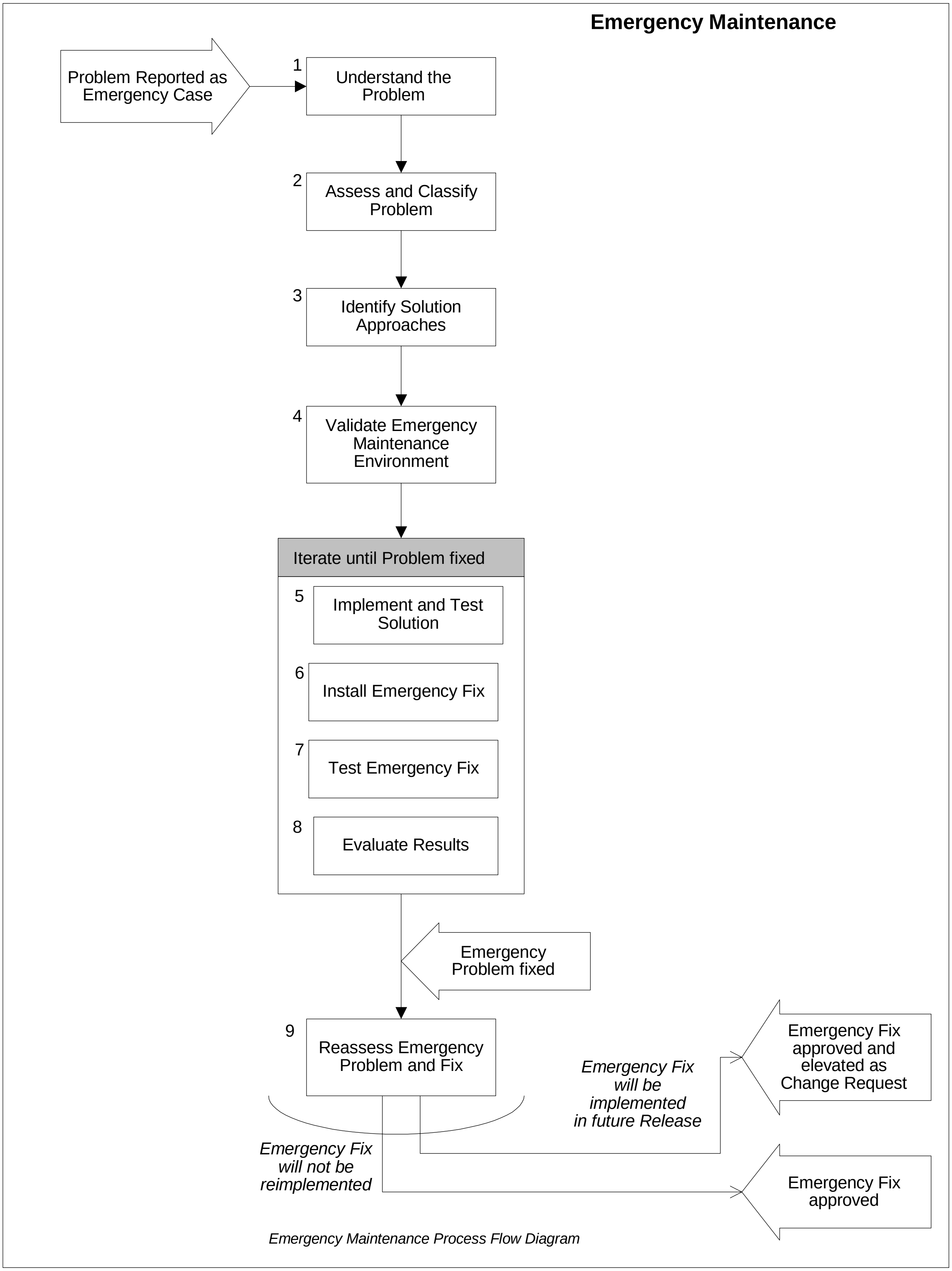

Emergency Maintenance
Problem Reported as Emergency Case
1
Understand the Problem
2
Assess and Classify Problem
3
Identify Solution Approaches
4
Validate Emergency Maintenance Environment
Iterate until Problem fixed
5
Implement and Test Solution
6
Install Emergency Fix
7
Test Emergency Fix
8
Evaluate Results
Emergency Problem fixed
9
Reassess Emergency Problem and Fix
Emergency Fix will be implemented in future Release
Emergency Fix approved and elevated as Change Request
Emergency Fix will not be reimplemented
Emergency Fix approved
Emergency Maintenance Process Flow Diagram

Figure 11–17. Emergency Maintenance, Process Flow Diagram

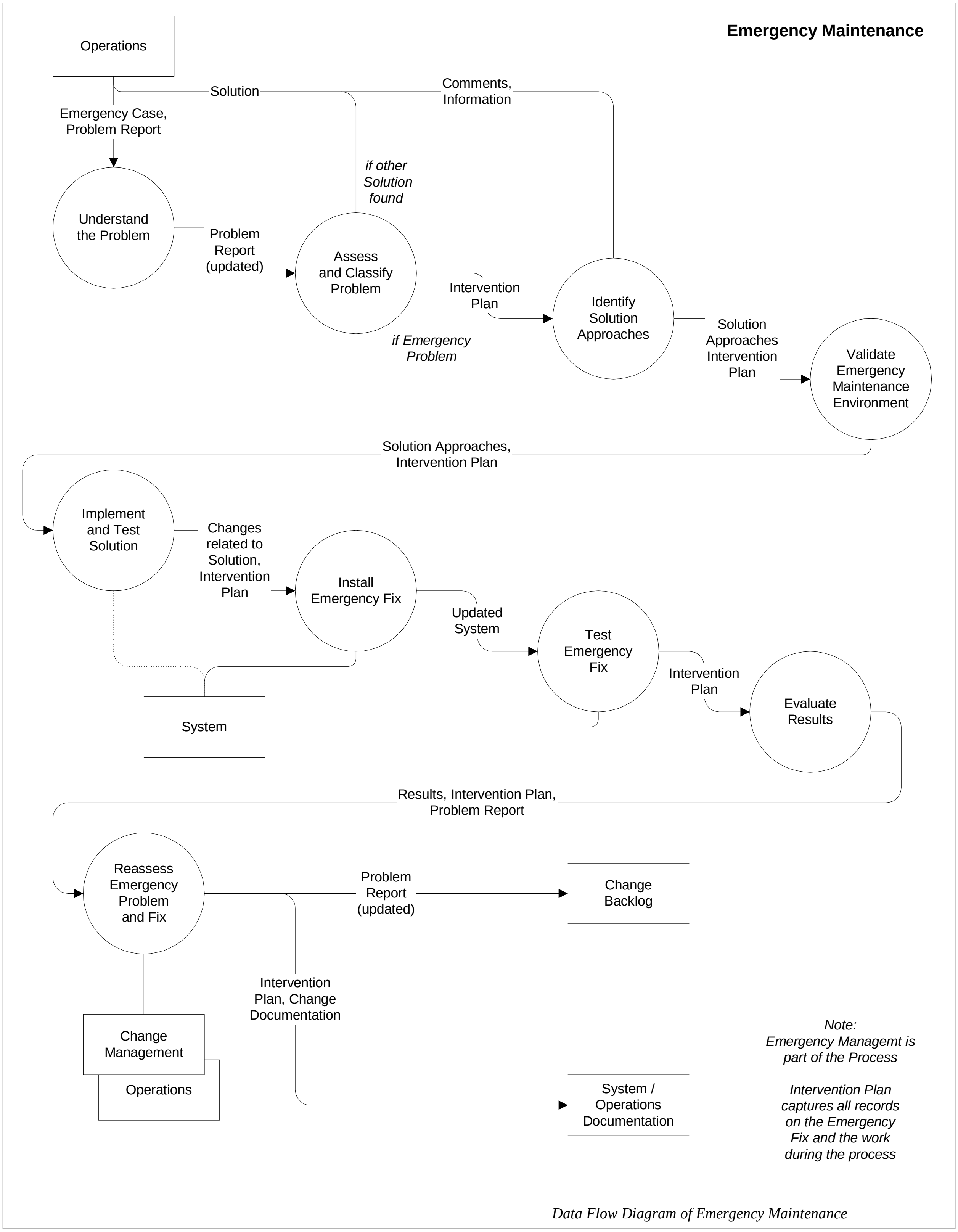

Emergency Maintenance
Operations
Solution
Comments, Information
Emergency Case, Problem Report
if other Solution found
Understand the Problem
Problem Report (updated)
Assess and Classify Problem
Intervention Plan
if Emergency Problem
Identify Solution Approaches
Solution Approaches Intervention Plan
Validate Emergency Maintenance Environment
Solution Approaches, Intervention Plan
Implement and Test Solution
Changes related to Solution, Intervention Plan
Install Emergency Fix
Updated System
Test Emergency Fix
Intervention Plan
Evaluate Results
System
Results, Intervention Plan, Problem Report
Reassess Emergency Problem and Fix
Problem Report (updated)
Change Backlog
Intervention Plan, Change Documentation
Change Management
Operations
System / Operations Documentation
Note: Emergency Managemt is part of the Process
Intervention Plan captures all records on the Emergency Fix and the work during the process
Data Flow Diagram of Emergency Maintenance

Figure 11–18. Emergency Maintenance, Data Flow Diagram

Table 11–9 associates the nine activities with the roles responsible for their execution.

Table 11–9. Emergency Maintenance, Activity-Role Matrix

Activities	Roles								
Accountable for Execution: Overall Project Manager, Technical Project Manager, Development Team Approval to Proceed: Business Advocate Provides Input: Business SME, Technical SME, Other Representatives or Stakeholders: CM, QA	OPM	TPM	DT	BA	BSME	TSME	OTHER	CM	QA
1. Understand the problem			P						
2. Assess and classify problem			P						
3. Identify solution approaches			P						
4. Validate emergency maintenance environment			P						
5. Implement and test solution		A	P						
6. Install emergency fix		A	P					R	
7. Test emergency fix			P						
8. Evaluate results of approach		R	P						
9. Reassess emergency problem and fix		P	P					R	

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

Figure 11–15, at the end of the previous section on routine (release-based) maintenance, depicts the flow of a software module through the various software libraries during emergency maintenance.

12. Component 7. Decommission the Solution

12.1 Purpose

The purpose of this component is to deactivate an operational system, which includes:

- Coordinate with the Records Management Branch
- Determine the impact on any other systems
- Remove the system from the operational environment
- Update the inventory

12.2 Roles and Responsibilities

The roles required to perform the activities in the Decommission the Solution component are shown in Table 12–1.

Table 12–1. Component 7 Roles and Responsibilities

Roles	Responsibilities
Overall Project Manager Technical Project Manager Development Team	Accountable for Execution
Business Advocate Executive Sponsor	Approval to Proceed
Business Project Manager Business SME Technical SME Other Representatives or Stakeholders • Quality Assurance Manager • Configuration Management Manager • Records Management	Provides Input

12.3 Entry Criteria

Any of the following events may trigger the initiation of Component 7:

- Service date
- Technology Development Plan Project selected
- Installation of a replacement technology item
- Records retention schedule expiration

12.4 Input, Activities, and Outputs

The inputs to this component are as follows:

- Project Management Plan
- Software Engineering Notebook
- Installed solution
- Operational Environment
- System Inventory
- Technical Reference Model
- GILS

Figure 12–1 illustrates the flow of the following activities:

1. Review and update plans and announce decommissioning:
2. Notify Records Management Branch to review records management requirements
3. Analyze system interfaces and document effect on any other systems
4. Obtain approval to remove the solution from the operational environment
5. Update documentation and records
6. Wait for confirmation and remove the system
7. Obtain final sign-off that decommissioning is complete

As the figure suggests, many of the activities can be performed in parallel. The activities are described in more detail in Section 12.7.

Figure 12–1 also summarizes the outputs of all Component 7 activities in the central data store.

Appendix C includes a summary description of all products of NRC Projects, indicates within which components each product is created and updated, and specifies which products are required.

The products, whose standard and form numbers are shown in the figure, are described in more detail in the companion volume *SDLCM Methodology Procedures, Standards, and Forms*.

12.5 Techniques and Tools

The activities of Component 7 are supported by the following techniques:

- Structured Facilitation
- Structured Walkthrough
- Peer Review

Refer to the current *SDLCM Methodology Tool Inventory* for the recommended set of tools.

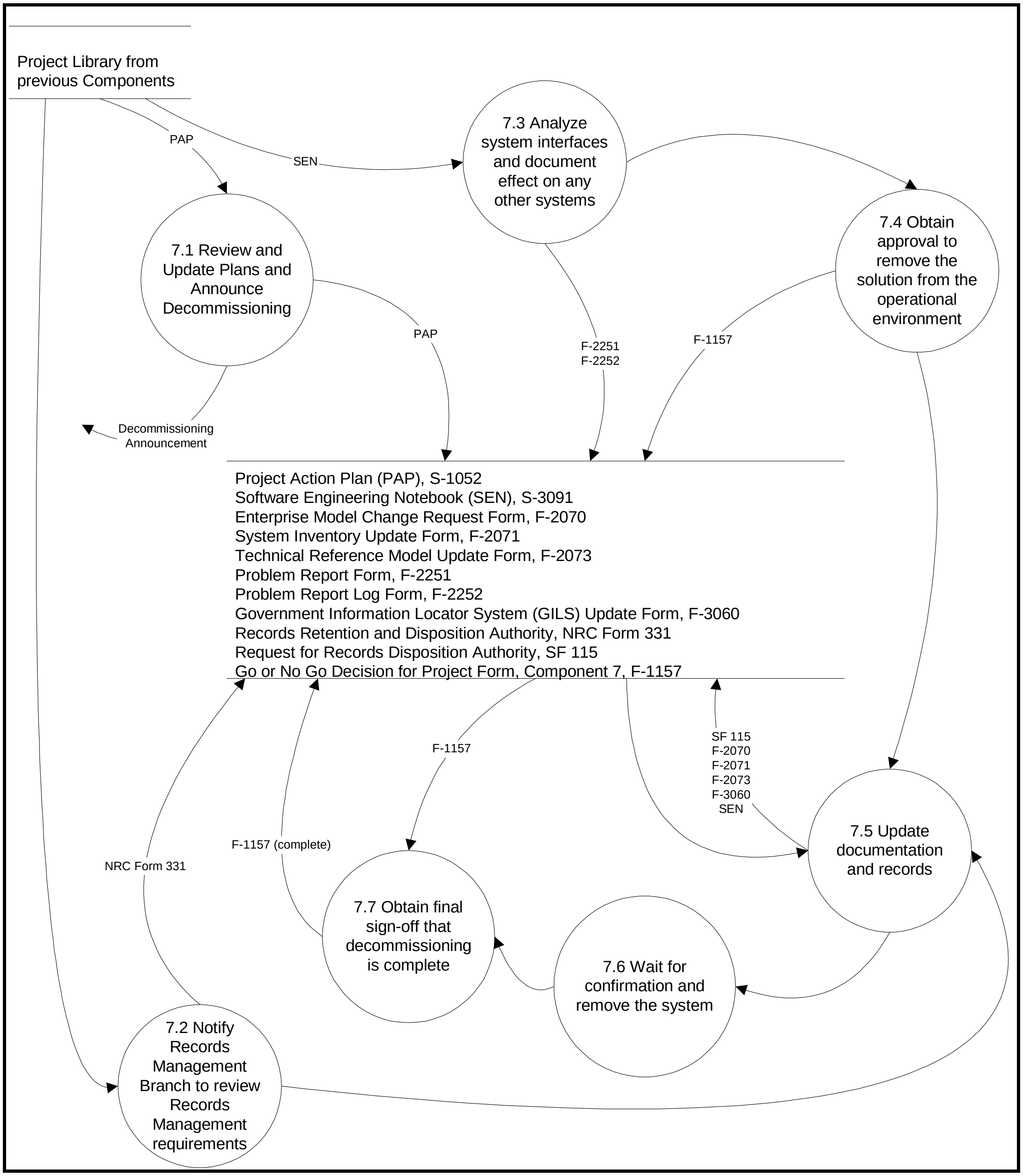

Project Library from previous Components
PAP
SEN
7.3 Analyze system interfaces and document effect on any other systems
7.4 Obtain approval to remove the solution from the operational environment
7.1 Review and Update Plans and Announce Decommissioning
PAP
F-2251
F-2252
F-1157
Decommissioning Announcement
Project Action Plan (PAP), S-1052
Software Engineering Notebook (SEN), S-3091
Enterprise Model Change Request Form, F-2070
System Inventory Update Form, F-2071
Technical Reference Model Update Form, F-2073
Problem Report Form, F-2251
Problem Report Log Form, F-2252
Government Information Locator System (GILS) Update Form, F-3060
Records Retention and Disposition Authority, NRC Form 331
Request for Records Disposition Authority, SF 115
Go or No Go Decision for Project Form, Component 7, F-1157
F-1157
SF 115
F-2070
F-2071
F-2073
F-3060
SEN
F-1157 (complete)
NRC Form 331
7.5 Update documentation and records
7.7 Obtain final sign-off that decommissioning is complete
7.6 Wait for confirmation and remove the system
7.2 Notify Records Management Branch to review Records Management requirements

Figure 12–1. Component 7

12.6 Exit Criteria

Component 7 is complete when:

- All critical products have been reviewed (structured walkthrough or peer review)
- Key products have been inspected by QA to conform to Project standards
- Products have been placed under CM control
- Information has been shared
- Issues have been resolved
- There is management approval that the decommissioning is complete

12.7 Component 7 Activity Details

The following pages provide detailed activities of Component 7, Decommission the Solution.

Activity 7.1: Review and Update Plans and Announce Decommissioning

See Figure 12–2.

- Review the Project Management Plan, and announce the decommissioning to the entire NRC community. It is important to prepare people to expect decommissioning activities. The system may have users of whom the owner is unaware.
- From the users' point of view, determine what will be done with the software, data, and documentation of the decommissioned system (for example, leave it in the inventory for possible future use, turn it over to a different organization, save it for a specified time period, destroy it). (Analysis of the legal and regulatory Records Management requirements may necessitate alternative disposition of the system.)
- Update the Project Management Plan.

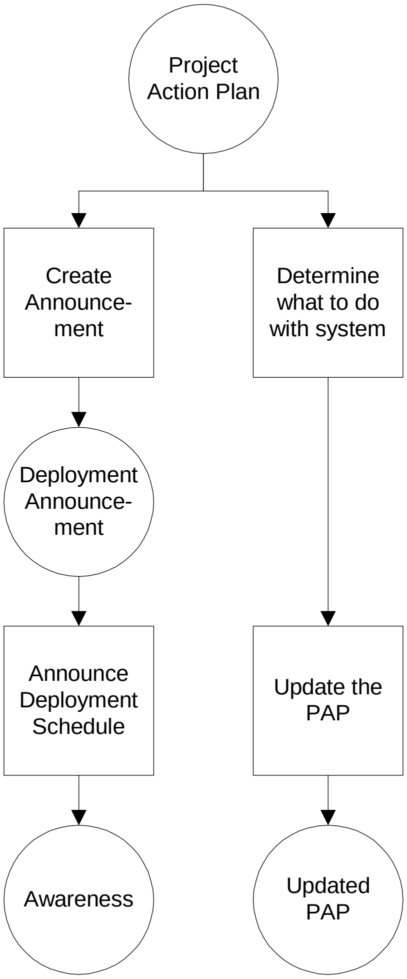


Figure 12–2. Review and Update Plans and Announce Decommissioning

Activity 7.2: Notify Records Management Branch to Review Records Management Requirements

See Figure 12–3.

1. Review copies of the previously submitted Records Management forms in the Project library:
   - Information System Description, NA Form 14028
   - Records Retention and Disposition Authority, NRC Form 331
   - Notification of Electronic Information System Design or Modification, NRC Form 616
   - Request for Records Disposition Authority, SF 115
2. Complete a new NRC Form 331 and deliver it to Records Management as notification that that an application system is about to be decommissioned.
3. After the form has been processed, combine the legal and regulatory requirements with the business and technical requirements (documented in Activity 7.3) and update the documentation in Activity 7.5.

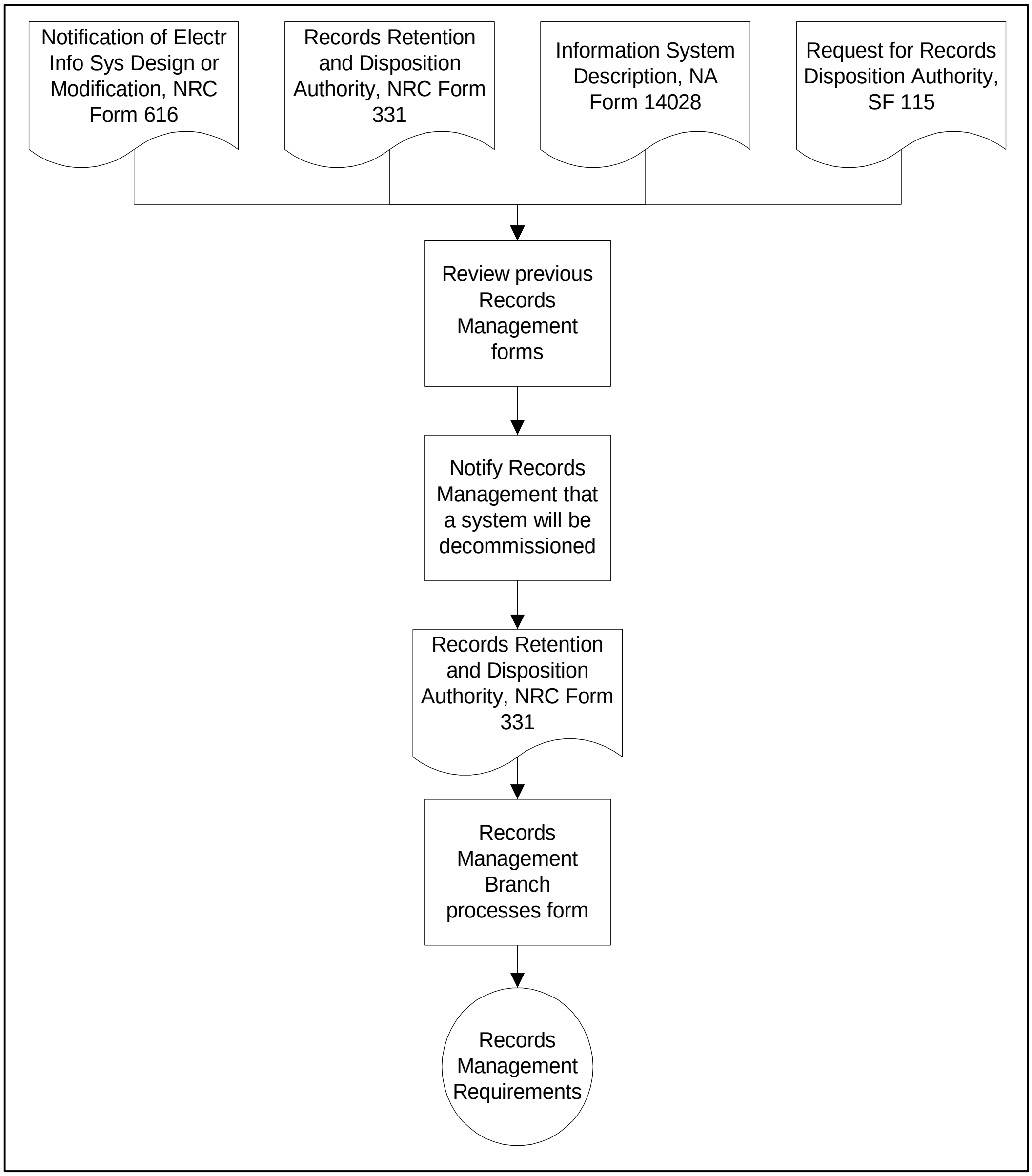

Notification of Electr Info Sys Design or Modification, NRC Form 616
Records Retention and Disposition Authority, NRC Form 331
Information System Description, NA Form 14028
Request for Records Disposition Authority, SF 115
Review previous Records Management forms
Notify Records Management that a system will be decommissioned
Records Retention and Disposition Authority, NRC Form 331
Records Management Branch processes form
Records Management Requirements

Figure 12–3. Notify Records Management Branch to Review Records Management Requirements

Activity 7.3: Analyze System Interfaces and Document Effect on Any Other Systems

See Figure 12–4. Using information in the Software Engineering Notebook,

1. Analyze the system to determine any external interfaces or dependencies.
2. If other application systems make calls to functionality provided by this system or make use of any data generated by this system, submit Problem Report(s) using Form F–2251 to the CCBs of the affected systems to document the fact that the dependencies must be removed. (NOTE: It is the responsibility of the owners of the affected systems to make the necessary changes or to request that this decommissioning activity be terminated. The decommissioning team must not initiate Activity 7.6 before receiving confirmation that all dependencies have been removed.)

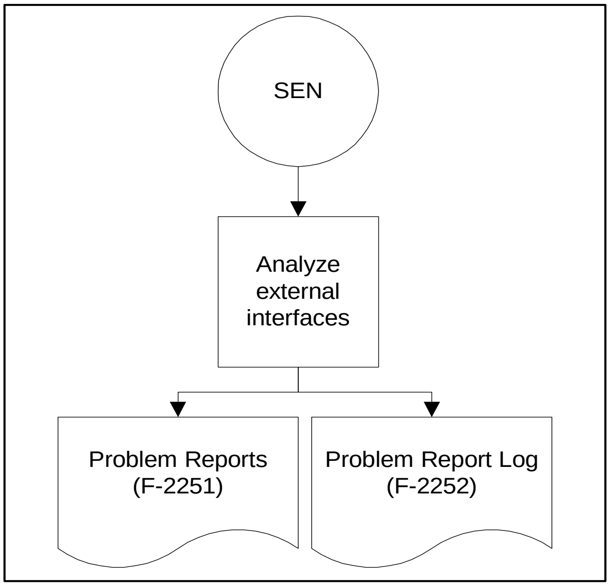


Figure 12–4. Analyze System Interfaces and Document Effect on Any Other Systems

Activity 7.4: Obtain Approval to Remove the Solution from the Operational Environment

See Figure 12–5. Complete the Go or No Go Decision for Project Form, Component 7, and submit it to the Business Advocate to get the user community's approval to remove the solution from the operational environment.

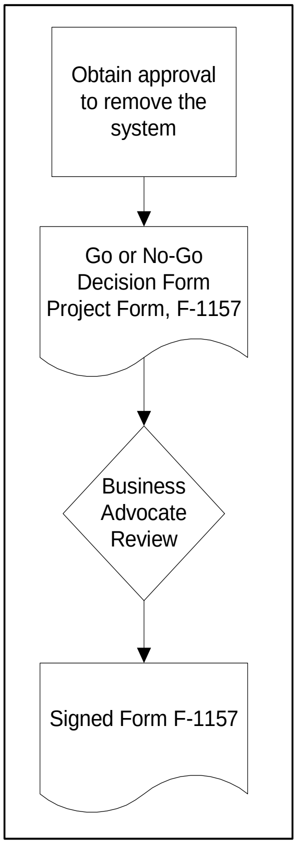

Obtain approval to remove the system
Go or No-Go Decision Form Project Form, F-1157
Business Advocate Review
Signed Form F-1157

Figure 12–5. Obtain Approval to Remove the Solution from the Operational Environment

Activity 7.5: Update Documentation and Records

See Figure 12–6. After the Records Management requirements are known and the interfaces have been analyzed:

1. Update the system documentation (SEN) unless everything will be destroyed immediately.
2. Update the Enterprise Model, System Inventory, TRM, and GILS.
3. Notify Records Management.
4. Update any affected operational policies and procedures to remove references to the application system about to be removed.

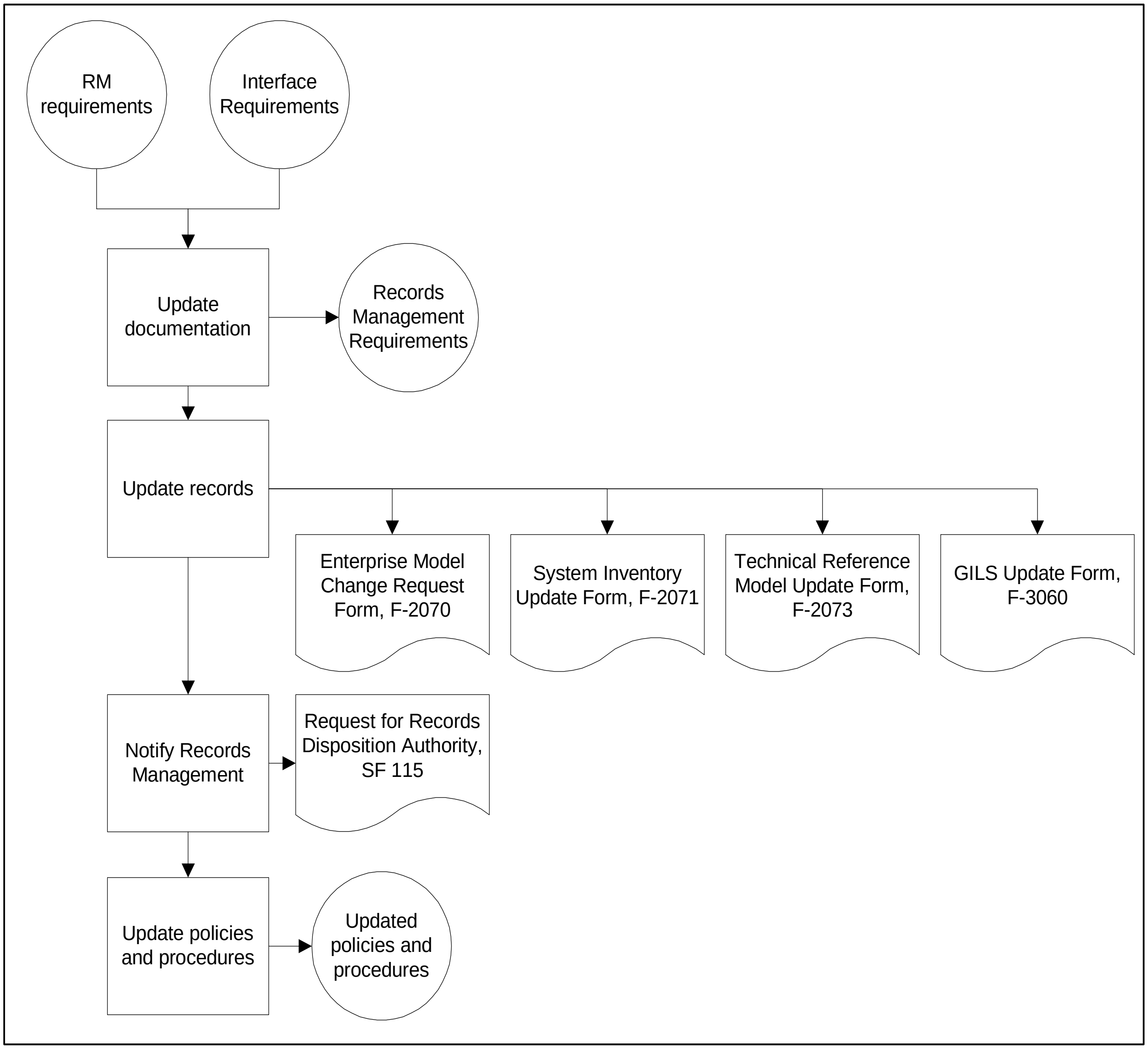

RM requirements
Interface Requirements
Update documentation
Records Management Requirements
Update records
Enterprise Model Change Request Form, F-2070
System Inventory Update Form, F-2071
Technical Reference Model Update Form, F-2073
GILS Update Form, F-3060
Notify Records Management
Request for Records Disposition Authority, SF 115
Update policies and procedures
Updated policies and procedures

Figure 12–6. Update Documentation and Records

Activity 7.6: Wait for Confirmation and Remove the System

See Figure 12–7.

1. Wait for confirmation that any affected systems have been re-deployed.
2. Using the plans for decommissioning documented in the PMP, remove the system from the operational environment. Archive or destroy related processes and data as documented in the PMP.

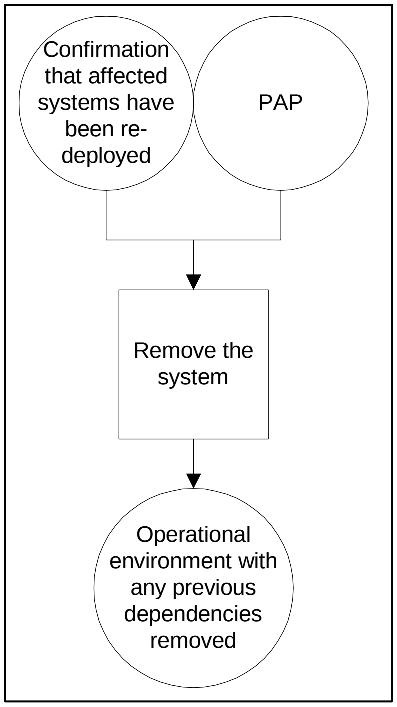

Confirmation that affected systems have been re-deployed
PAP
Remove the system
Operational environment with any previous dependencies removed

Figure 12–7. Wait for Confirmation and Remove the System

Activity 7.7: Obtain Final Sign-Off That Decommissioning Is Complete

See Figure 12–8. After the system has been removed from the environment and all required documentation has been completed, obtain the final sign-off from the Executive Sponsor that the decommissioning is complete.

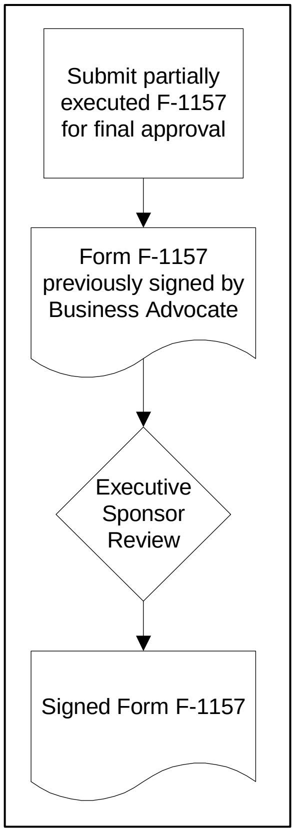

Submit partially executed F-1157 for final approval
Form F-1157 previously signed by Business Advocate
Executive Sponsor Review
Signed Form F-1157

Figure 12–8. Obtain Final Sign-Off That Decommissioning Is Complete

Appendix A. Maintaining the SDLCM Methodology

The SDLCM Methodology is the approach to doing business at the NRC. That approach is described by a set of documents, including this handbook and the companion volume of procedures, standards, and forms.

As discussed in Section 2.4, every Project provides an opportunity to improve the process defined by the SDLCM Methodology.

SDLCM Methodology Procedure P–9001 (SDLCM Methodology Change) defines the mechanism for reporting deficiencies—or suggesting improvements—in the methodology or in the documents describing the methodology. To request a change to the SDLCM Methodology or to any part of its documentation set, follow the steps specified in Procedure P–9001.

SDLCM Methodology Form F–9001 (SDLCM Methodology Change Request Form) is the vehicle for documenting any change.

Appendix B. Roles and Responsibilities

This appendix provides a detailed summary of the responsibilities of the various Project roles identified in the SDLCM Methodology.

Roles	Responsibilities
Executive Sponsor	• Being the champion of the business change Project and the initial source for determining its scope and objectives • Being the chief decision-maker responsible for the business program and being ultimately responsible for the decision to have the process automated or the system upgraded • Being the senior business official (typically an SES) who approves the Project and authorizes resources • Having approval authority for budget • Being involved in scoping and high-level decision-making for the Project • Empowering Business Advocates to participate fully in the Project • Approving requirements definition, authorizing resources, approving design, authorizing deployment, and decommissioning • Providing funding and ensuring long-term support for implementation • Using political influence and corporate knowledge to ensure Project success • Approving (go or no-go decisions) at milestones identified in action plans • Monitoring Project progress • Resolving sensitive or critical issues as needed • Making policy decisions • Communicating Project activities and successes throughout the organization • Communicating to the team external events and situations that may affect the Project on a timely basis • Clearing calendars of Project team members as needed

Roles	Responsibilities
Overall Project Manager	• Planning the Project, allocating and directing the staff and other resources to accomplish Project tasks, and maintaining control over the Project • Being responsible for Project coordination and day-to-day contacts • Having high level knowledge of business and technology • Coordinating and developing the business *and* technical aspects of the Project • Being responsible for the deliverables, cost, schedule, and quality of the Project • Providing Project team leadership to business and technical staff • Coordinating business representation and IS activities (through Project management plan which includes any significant milestones from the Software Development Plan and all non-software development milestones). • Removing obstacles to Project success • Coaching, developing, and supporting Project team members • Ensuring all team members have appropriate training and experience • Maintaining totally integrated control of the Project • Managing all the business aspects of the Project including manual process delivery products • Providing all briefing and status reporting information to all levels of interested staff and management. • Perhaps also playing the roles of the Business Project Manager or the Technical Project Manager, as determined on a Project-by-Project basis • Recommending go or no-go decisions at each milestone for Executive Sponsor approval
Business Project Manager	• Sharing responsibility for Project coordination and day-to-day contacts with the Overall Project Manager • Having high level knowledge of process and requirements • Representing the sponsoring office (typically non-SES) • Directing and overseeing day-to-day Project activities affecting the business view of the Project • Coordinating and developing the business aspects of the Project • Reporting progress and involving appropriate customer and business experts • Recommending go or no-go decision at each milestone for Executive Sponsor approval • Managing business office resources in the design, engineering, and deployment of the solution • Working with the Technical Project Manager • Developing, documenting and selling policy and practice changes • Perhaps also playing the role of the Overall Project Manager, as determined on a Project-by-Project basis • Develop acceptance tests, user tests, business knowledge tests and training materials, and Systems Operations Concept

Roles	Responsibilities
Technical Project Manager	• Being responsible for technical Project coordination and development of the Software Development Plan • Having high-level knowledge of technology, tools, and methodology • Being the technical representative (typically non-SES) • Having responsibility for all IT-related aspects of the Project • Managing all technical aspects of the Project • Marshaling the Development Team • Working with the Business Project Manager and Overall Project Manager • Managing all technical development of the automated processes of the Project and associated products • Managing staff and contractors involved in Project • Controlling the development schedule for automated processes • Managing developer resources and the use of the methodology • Coordinating development activities to conform to business requirements • Ensuring that the SDLCM Methodology is followed • Perhaps also playing the role of the Overall Project Manager, as determined on a Project-by-Project basis

Roles	Responsibilities
Business Advocate	• Being the primary beneficiary or stakeholder • Being the business leader whose business area is affected most by the Project and who has the lead role in the direction and use of the application outcome • Driving system requirements • Holding a key role in the business area where the system is being implemented and perhaps being: ⇨ The customer who has a vested interest in the Project (that is, a real user) ⇨ The functional area manager responsible for the business area where the solution is required ⇨ The principal user responsible for success of the Project • Keeping the vision for the business needs of the Project • Being a visionary who identifies problems and recognizes the need for improvement in a business area • Seeking authorization and approval from the Executive Sponsor • Being an enthusiast, a sales person, a resource identifier, and a business verifier • Overcoming Project problems in all areas (resources, scoping, design, selling, installation) • Selling the solution upward to executive management and executive sponsorship • Selling the business need downward and outward to all involved with the Project (equivalent of private industry program or matrix management) • Ensuring that all business requirements have been addressed by the solution • Reviews and recommends approval of all Project plans and products/deliverables • Identifying and resolving critical issues • Obtaining approval for new business policies and practices to support the new solution • Participating in the implementation of the solution within his or her own business area • Working closely with development team to ensure compliance with business area requirements • Testing and reporting discrepancies

Roles	Responsibilities
Software Development Team Individual team roles may include: • Data Modeler • Database Administrator • Knowledge Coordinator • Analyst • JAD Facilitator • Prototype Developer • Designer • Coder • Tester • Configuration Manager • QA representative	• Doing the detailed work on one or more aspects of the system (definition, design, engineering, deployment, service or decommissioning) • Building applications • Developing prototypes • Gathering technical alternatives • Refining, designing, developing, and deploying the system • Gathering, validating, and testing against user and technical requirements • Testing (and resolving issues) • Documenting the requirements, design, software, and operational and maintenance processes • Developing and obtaining approval for all Project products • Developing user guides • Controlling Project baselines and changes to those baselines • Ensuring that quality is built into the system • Following SDLCM Methodology guidance
Development Team Individual team roles may include: • Technical Project Manager • Technical Subject Matter Expert • Software Development Team	• Overseeing all aspects of the system effort • Evaluating technical alternatives • Overseeing system deployment • Ensuring user and technical requirements are met by the system • Obtaining approval for all Project products • Ensuring that quality is built into the system • Following SDLCM Methodology guidance
Business Subject Matter Expert (SME)	• Advising the Business Advocates concerning business area requirements • Providing detailed functional expertise input and guidance throughout Project life cycle • Clarifying business requirements • Providing input to prototype refinements • Providing business area information not available from Business Advocates • Participating in the development by providing the business perspective • Providing input to validate feasibility of prototype or design • Perhaps being an industry expert, a business function expert, or a business process expert

Roles	Responsibilities
Technical Subject Matter Expert (SME)	• Being an authority who has specialized knowledge of the technical side of the Project • Providing technical expertise to the Development Team • Being infrastructure experts, component hardware or software experts, application package experts, or product and technology experts • Working closely with the Development Team to ensure that the technical solution conforms to business requirements • Providing detailed technical expertise • Identifying potential fixes • Providing input that defines or clarifies technical requirements • Advising the Development Team on alternative designs, development approaches, applicable new technologies, etc.
Other Representatives or Stakeholders	• Representing the business needs of non-lead office(s) significantly affected by the system being developed or affected by the outcome of the solution • Providing input and perspective to Project plans and deliverables

Appendix C. Products of Projects

This appendix identifies and summarizes all products of Projects as specified by NRC's SDLCM Methodology and specifies which products are required. Use this appendix together with Appendix D, which summarizes the activities and identifies the products of each activity, as a cross-reference to the component descriptions in the main body of this handbook.

Section 2 summarizes the products of NRC Projects, organized by the component of the methodology within which the product is initially created.

Section 2 addresses legacy systems.

C.1 Products of Projects by SDLCM Methodology Component

The tables in this section summarize the products according to the components in which they are initially created. For a convenient listing of all products ordered by product identification numbers, see the Table of Contents of the companion volume of *SDLCM Methodology Procedures, Standards, and Forms*. The standards and forms in that volume represent the products of Projects.

The SDLCM Methodology distinguishes among activities associated with (1) new development, (2) enhancement, and (3) maintenance of application systems. See Chapter 2, "Methodology Overview," and Chapter 11, "Component 6. Service the Solution," for more information.

- Tables C–1 through C–5 (products of Components 1 through 5) pertain to new development and enhancement Projects.
- Table C–6 (products of Component 6) pertains to operational systems that have been deployed and are under maintenance.
- Table C–7 (products of Component 7) pertains to any operational system that is being decommissioned.

When performing *maintenance* (Component 6), update only the system documentation products that are needed to support further system maintenance. For example, if the software or system tests are modified, then update the Test Plan; if the user interface or functionality is modified, then update the User Guide. Do not update any system documentation products that are not affected by the maintenance change. See also Section C.2.

Table C–1. Products Created within Component 1

Product Name	Created in Component	Updated in Components	Product Summary	Required Product?
Project Charter (S–1051)	1		The Project charter is a management document used to initiate an NRC Project that will be developed 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.	Required for new development Projects. Not required for enhancement Projects.
System Requirements Specification (S–3051)	1	3,4,6	The SRS clarifies the scope of the Project. It contains the system functional requirements, data requirements, an assessment of the current system (if any), an analysis of alternative approaches for satisfying the requirements, and a system operations concept.	Required
Current System Assessment Document (S–3052)	1		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 System Requirements Specification.	Required
Project Management Plan (S–1052)	1	2,3,4,5,6,7	The PMP communicates to management, the customer, and Project members the overall plan for performing and managing the Project from start to end.	Required
Software Development Plan (S-1057)	1	2,3,4,5,6,7	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 software requirements have been identified.	Required
System Operations Concept (S-3053)	1	2,3	Initially developed by the Overall Project Manager, Business Advocates, and Business Subject Matter Experts as a high level concept of the envisioned system for use in validating completeness of requirements list and by designers in developing process models and data models.	Required
Alternatives Analysis Document (S-3054)	1	2,3	Used by those persons who are responsible for the evaluation of alternative approaches to addressing the options and associated costs and risks for delivering new or enhanced application systems or correcting problems with existing application systems.	Required

Product Name	**Created in Component**	**Updated in Components**	**Product Summary**	**Required Product?**
Quality Assurance Plan (S-2001)	1	2,3	The Quality Assurance Plan (QAP) is a management document, based on the Project Management Plan (PMP), that defines the quality role and responsibilities during development and/or maintenance for a specific Project. This information includes identification of quality resources, activities and processes that are to be implemented for the identified Project.	Required
Configuration Management Plan (S-3501)	1	2,3	The Configuration Management Plan (CMP) is a management document that identifies the organization that performs Configuration Management (CM) for the Project/Task. It is typically produced in a high level form during the early components of the SDLCMM process and is defined at a lower level of detail by the end of Component 3.	Required
Tactical Integration Plan (S–5051)	1	2,3,4,5,6	The TIP serves as notification of the intended deployment of a system or application and provides the time frame for scheduled operation. It provides NRC management with the information necessary to (1) ensure that the level of planning is sufficient to proceed with the system deployment described, (2) assess the impact on other components of the NRC Enterprise Model, and (3) confirm the adequacy of the schedule and budget to complete the deployment and initial operation.	Required
Support Resource Acquisition Request and Commitment Form (F–1061)	1,6		Use this form, if needed, to identify and request personnel and other resources.	Required only if needed
Environment Change Request Form (F–1601)	1,6		Use this form, if needed, to identify and request new or upgraded technology (including tools) that will change the NRC computing environment.	Required only if needed
Notification of Electronic Information System Design or Modification (NRC Form 616)	1,6		Use this form to notify the Records Management Branch that a system is being designed or modified. RM will assign a representative to act as its interface with the Project Manager.	Required
Records Retention and Disposition Authority (NRC Form 331)	1,7		Use this form to notify the Records Management Branch of the potential need to establish or change the disposition of official records.	Required
Information System Description (NA Form 14028)	1		Use this form to provide a description of the system to the Records Management Branch.	Required
Request for Records Disposition Authority (SF 115)	1,7		Use this form to propose a disposition of records. RM will submit the request to NARA.	Required

Product Name	**Created in Component**	**Updated in Components**	**Product Summary**	**Required Product?**
Data Models (S–3151)	1	3,6	Data models may be included in the SRS, the Logical Design Document, and the Physical Design Document	Required
Process Models (S–3161)	1	3,6	Process models may be included in the SRS, the Logical Design Document, and the Physical Design Document	Required
Context Diagrams (S–3162)	1	3,6	A context diagram must be included in the SRS and may be updated and included in the Logical Design Document	Required
Data Flow Diagrams (S–3163)	1	3,6	Data flow diagrams may be included in the SRS, the Logical Design Document, and the Physical Design Document	Required
Data Dictionary (S–3351)	1	3,6	The data dictionary is a mechanism for defining a system's data elements. Information about the data dictionary may be included in the SRS, the Logical Design Document, and the Physical Design Document	Required
SDLCM Methodology Deviation or Waiver Form (F–2010)	1		Use this form to request a deviation or waiver from a requirement of the SDLCM Methodology. (Deviation: fulfill the requirement with modifications. Waiver: eliminate the requirement.)	Required only if needed
Enterprise Model Change Request Form (F–2070)	1,3,7		Use this form to report a change to the NRC Enterprise Model	Required
Go or No GO Decision for Project Form, Component 1 (F–1151)	1		Use this form to request approval to proceed with the activities of Component 2, Acquire Support Resources.	Required

Table C–2. Products Created or Updated within Component 2

Product Name	Created in Component	Updated in Components	Product Summary	Required Product?
Project Management Plan (S–1052)	1	2,3,4,5,6,7	(See Component 1.)	Required
Tactical Integration Plan (S–5051)	1	2,3,4,5,6	(See Component 1.)	Required
Development and Maintenance Environment Products Installation Plan (S–1055)	2	6	The Development and Maintenance Environment Products Installation Plan provides a detailed description of the activities involved in the installation of all resources required to design, engineer, and maintain the system. It defines responsibilities, schedules, risks, and risk mitigation approaches.	Required only if needed
Statement of Work (S–1053)	2		The SOW is the Project manager's description of the resources required from the 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.	Required only if needed
Support Resource Acquisition Strategy Form (F–1062)	2		Use this form to document the strategy for acquiring the needed resources.	Required only if needed
Developer Training Requirements Form (F–7051)	2		Use this form to document the training requirements for Project personnel.	Required only if needed
Fully Staffed and Trained Team Form (F–1181)	2		Use this form to document the roles of the Project personnel and to document that required training has been completed.	Required
Go or No GO Decision for Project Form, Component 2 (F–1152)	2		Use this form to request approval to proceed with the activities of Component 3, Design the Solution.	Required

Table C–3. Products Created or Updated within Component 3

Product Name	Created in Component	Updated in Components	Product Summary	Required Product?
Project Management Plan (S–1052)	1	2,3,4,5,6,7	(See Component 1.)	Required
Software Development Plan (S-1057)	1	2,3,4,5,6,7	(See Component 1.)	Required
System Requirements Specification (S–3051)	1	3,4,6	(See Component 1.)	Required
Logical Design Document (S–3171)	3	4,6	The Logical Design Document presents the system architecture at a level of detail sufficient to begin detailed physical design activities. It translates the system's requirements, contained in the SRS, 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.	Required
Physical Design Document (S–3172)	3	4,6	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, a Data Dictionary, and the screen prototypes.	Required
Tactical Integration Plan (S–5051)	1	2,3,4,5,6	(See Component 1.)	Required
Products Installation and Integration Plan (S–5052)	3	4,5,6	The Products Installation and Integration Plan provides a detailed description of the activities involved in the installation and integration of equipment resources (Commercial or Government off-the-shelf (COTS or GOTS) hardware and software and custom-developed software) required for the deployment of the solution system. It defines responsibilities, schedules, risks, and risk mitigation approaches.	Required if not included in update of TIP
Solution Integration Plan (S–5053)	3	4,5,6	The Solution Integration Plan provides a detailed description of the activities involved in the integrating the hardware and software configuration items (CIs) of the solution system. It also specifies roles and responsibilities and the integration schedule, and identifies any risks, and risk mitigation approaches applicable to the integration process.	Required if not included in update of TIP
Other Systems Integration Plan (S–5054)	3	4,5,6	The Other Systems Integration Plan provides a detailed description of the activities involved in integrating the solution system with other NRC systems, both legacy and new. It defines responsibilities, schedules, risks, and risk mitigation approaches.	Required if not included in update of TIP

Product Name	Created in Component	Updated in Components	Product Summary	Required Product?
External Systems Interface Diagrams (S–3164)	3		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 illustrates process, external agents, external interfaces, and shared external databases. The diagram may be included in the LDD and PDD, if needed.	Required in LDD and PDD if there are interfaces to external systems
Other Integration Issues Plan (S–5055)	3	4,5,6	The Other Integration Issues Plan provides a detailed description of the activities involved in accomplishing any integration issues not covered by other plans. It defines responsibilities, schedules, risks, and risk mitigation approaches.	May be included in the Tactical Integration Plan (TIP) for integration issues not covered by any other integration plan.
Conversion Plan (S–1054)	3	4,5,6	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.	Required if not included in update of TIP
Security Controls (S–1056)	3	4,5,6	Specifies the set of security controls to be included in the developed system.	Required if not included in update of TIP
Integrated Education, Training, and Reference Materials (S–7052)	3	6	Summarizes the results of the training design activities into an organized and logically flowing deliverable.	Required
Data Models (S–3151)	1	3,6	(See Component 1.)	Required
Process Models (S–3161)	1	3,6	(See Component 1.)	Required
Context Diagrams (S–3162)	1	3,6	(See Component 1.)	Required
Data Flow Diagrams (S–3163)	1	3,6	(See Component 1.)	Required
Data Dictionary (S–3351)	1	3,6	(See Component 1.)	Required
Software Engineering Notebook (S-3091)	3	4,5,6	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.	Required

Product Name	Created in Component	Updated in Components	Product Summary	Required Product?
Test Plan (S-5151)	3	4,5,6	The Test Plan (TP) identifies the informal and formal testing that is planned in the Project/Task at each of the Unit, Integration, Qualification and Acceptance Test levels, the test scenarios to be used, and the roles and responsibilities for testing.	Required
Enterprise Model Change Request Form (F–2070)	1,3,7		(See Component 1.)	Required
Technical Reference Model Update Form (F–2073)	3,5,7		Use this form to report a change to the NRC Technical Reference Model	Required
Go or No GO Decision for Project Form, Component 3 (F–1153)	3		Use this form to request approval to proceed with the activities of Component 4, Engineer the Solution.	Required

Table C–4. Products Created or Updated within Component 4

Product Name	Created in Component	Updated in Components	Product Summary	Required Product?
Project Management Plan (S–1052)	1	2,3,4,5,6,7	(See Component 1.)	Required
Software Development Plan (S-1057)	1	2,3,4,5,6,7	(See Component 1.)	Required
System Requirements Specification (S–3051)	1	3,4,6	(See Component 1.)	Required
System Operations Concept (S-3053)	1	4	(See Component 1.)	Required
Alternatives Analysis Document (S-3054)	1	4	(See Component 1.)	Required
Tactical Integration Plan (S–5051)	1	2,3,4,5,6	(See Component 1.)	Required
External Systems Interface Diagram (S-3164)	3	4	(See Component 3.)	Required
Logical Design Document (S–3171)	3	4,6	(See Component 3.)	Required
Physical Design Document (S–3172)	3	4,6	(See Component 3.)	Required
Test Plan (S–5151)	3	4,5,6	(See Component 3.)	Required
Other Integration Issues Plan (S–5055)	3	4,5,6	(See Component 3.)	Required if not included in update of TIP
Conversion Plan (S–1054)	3	4,5,6	(See Component 3.)	Required if not included in update of TIP

Product Name	Created in Component	Updated in Components	Product Summary	Required Product?
Security Controls (S–1056)	3	4,5,6	(See Component 3.)	Required if not included in update of TIP
Solution Integration Plan (S-5053)	3	4	(See Component 3.)	Required if not included in update of TIP
Other Systems Integration Plan (S-5054)	3	4	(See Component 3.)	Required if not included in update of TIP
Installation Instructions (S–5252)	4	5, 6	The Installation Instructions document provides a detailed description of the activities involved in the installation of a new or enhanced NRC system and the equipment resources, both hardware and software, required for its support in the production environment.	Required if not included in update of TIP
Operational Support Guide (S–6151)	4	6	The Operational Support Guide provides guidance to operators and system support personnel on how to support users by performing specific tasks in a timely manner. It includes not only servicing users on an event-driven basis, but also performing some tasks periodically.	Required
User Training and Orientation Plan (S–7053)	4	6	The User Training and Orientation Plan provides a detailed plan for assessing the skills of users, providing training and orientation to enable users to operate the new system, specifying the budget and training schedule, and assessing the effectiveness of the training program.	Required
User Guide (S–6051)	4	6	The User Guide provides guidance on how to use the system effectively and efficiently.	Required
On-Line Help System and Tutorials (S–6052)	4	6	The On-Line Help System and Tutorials document is a brief overview of the on-line help and tutorials that are available.	Required
Software Engineering Notebook (S–3091)	3	4,5,6,7	(See Component 3.)	Required
Network Integration Diagrams (S–5056)	4		A set of network diagrams depicts the network support required by the logical design of the system. The form of the diagrams will be determined by the size and complexity of the system being developed or enhanced. The diagrams are included in the TIP.	Required in the TIP if the system runs on a network

Product Name	Created in Component	Updated in Components	Product Summary	Required Product?
Target Users Capability Upgrade Request Form (F–6054)	4,6		Use this form to request capability upgrades for targeted users.	Required only if needed
Change Log Form (F–2561)	4,5,6		The configuration management office uses this form to maintain a record of all change proposals.	Required only if Change Proposals have been submitted
Change Proposal Form (F–2502)	4,5,6		Use this form to request a change to a baselined system.	Required only if changes are needed
Go or No GO Decision for Project Form, Component 4 (F–1154)	4		Use this form to request approval to proceed with the activities of Component 5, Deploy the Solution.	Required

Table C–5. Products Created or Updated within Component 5

Product Name	Created in Component	Updated in Components	Product Summary	Required Product?
Project Management Plan (S–1052)	1	2,3,4,5,6,7	(See Component 1.)	Required
Tactical Integration Plan (S–5051)	1	2,3,4,5,6	(See Component 1.)	Required
Test Plan (S-5151)	3	4,5,6	(See Component 3.)	Required
Other Integration Issues Plan (S–5055)	3	4,5,6	(See Component 3.)	Required if not included in update of TIP
Change Proposal Form (F–2502)	4,5,6		(See Component 4.)	Required only if changes are needed
Problem Report Form (F–2251)	5,6,7		Use this form to report a problem with a deployed system.	Required only if a problem is identified
Problem Report Log Form (F–2252)	5,6,7		The configuration management office uses this form to maintain a record of all Problem Reports.	Required only if Problem Reports have been submitted
Software Engineering Notebook (S–3091)	3	4,5,6,7	(See Component 3.)	Required
System Inventory Update Form (F–2071)	5,7		Use this form to submit an update to the System Inventory	Required
Technical Reference Model Update Form (F–2073)	3,5,7		(See Component 3.)	Required
Government Information Locator System (GILS) Update Form (F–3060)	5,7		Use this form to submit an update to the GILS.	Required
Installation Instructions (S–5252)	4	5,6	(See Component 4.)	Required if not included in update of TIP
Installation Report Form (F-5254)	5,6		Use this form to report the results of installing the system.	Required only if needed
Rollout Report Form (F–5255)	5,6		Use this form to report on the readiness for the system to be moved to operational status.	Required only if needed
Performance Report Form (F–5256)	5		Use this form to report that the system is ready for acceptance testing.	Required only if needed

Product Name	Created in Component	Updated in Components	Product Summary	Required Product?
Deployment Recommendations for Fix Form (F–5257)	5		Use this form to recommend changes that should be made prior to acceptance testing.	Required only if needed
Customer Satisfaction Form (F–6055)	5,6		Use this form to document customer satisfaction with the system.	Required
Final User Signoff Form (F–6056)	5,6		Use this form as a delivery checklist and to document user acceptance.	Required
Trained Users Form (F–7151)	5		Use this form to document that the users have been trained to operate the system.	Required only if needed
Go or No GO Decision for Project Form, Component 5 (F–1155)	5		Use this form to request approval to proceed with the activities of Component 6, Service the Solution.	Required

Table C–6. Products Created or Updated within Component 6

Product Name	Created in Component	Updated in Components	Product Summary	Required Product?
Project Management Plan (S–1052)	1	2,3,4,5,6,7	(See Component 1.)	Required
Software Development Plan (S-1057)	1	2,3,4,5,6,7	(See Component 1.)	Required
System Requirements Specification (S–3051)	1	3,4,6	(See Component 1.)	Required
Tactical Integration Plan (S–5051)	1	2,3,4,5,6	(See Component 1.)	Required
Products Installation and Integration Plan (S–5052)	3	4,5,6	(See Component 3.)	Required if not included in update of TIP
Solution Integration Plan (S–5053)	3	4,5,6	(See Component 3.)	Required if not included in update of TIP
Other Systems Integration Plan (S–5054)	3	4,5,6	(See Component 3.)	Required if not included in update of TIP
Other Integration Issues Plan (S–5055)	3	4,5,6	(See Component 3.)	Required if not included in update of TIP
Conversion Plan (S–1054)	3	4,5,6	(See Component 3.)	Required if not included in update of TIP
Security Plan (S–1056)	3	4,5,6	(See Component 3.)	Required if not included in update of TIP
Installation Instructions (S–5252)	4	5,6	(See Component 4.)	Required if not included in update of TIP
Development and Maintenance Environment Products Installation Plan (S–1055)	2	6	(See Component 2.)	Required
Logical Design Document (S–3171)	3	4,6	(See Component 3.)	Required

Product Name	Created in Component	Updated in Components	Product Summary	Required Product?
Physical Design Document (S–3172)	3	4,6	(See Component 3.)	Required
Test Plan (S–5151)	3	4,6	(See Component 3.)	Required
Integrated Education, Training, and Reference Materials (S–7052)	3	6	(See Component 3.)	Required
Software Engineering Notebook (S–3091)	3	4,5,6,7	(See Component 3.)	Required
Operational Support Guide (S–6151)	4	6	(See Component 4.)	Required
User Training and Orientation Plan (S–7053)	4	6	(See Component 4.)	Required
User Guide (S–6051)	4	6	(See Component 4.)	Required
On-Line Help System and Tutorials (S–6052)	4	6	(See Component 4.)	Required
Data Models (S–3151)	1	3,6	(See Component 1.)	Required
Process Models (S–3161)	1	3,6	(See Component 1.)	Required
Context Diagrams (S–3162)	1	3,6	(See Component 1.)	Required
Data Flow Diagrams (S–3163)	1	3,6	(See Component 1.)	Required
Data Dictionary (S–3351)	1	3,6	(See Component 1.)	Required
Support Resource Acquisition Request and Commitment Form (F–1061)	1,6		(See Component 1.)	Required only if needed
Environment Change Request Form (F–1601)	1,6		(See Component 1.)	Required only if needed

Product Name	Created in Component	Updated in Components	Product Summary	Required Product?
Notification of Electronic Information System Design or Modification (NRC Form 616)	1,6		(See Component 1.)	Required
Solution Modules Ready for Deployment Form (F–4051)	4,6		(See Component 4.)	Required only if needed
Solution Rollout Support Ready for Deployment Form (F–4052)	4,6		(See Component 4.)	Required only if needed
Change Log Form (F–2561)	4,6		(See Component 4.)	Required only if Change Proposals have been submitted
Change Proposal Form (F–2502)	4,5,6		(See Component 4.)	Required only if changes are needed
Installation Report Form (F-5254)	5,6		(See Component 5.)	Required only if needed
Problem Report Form (F–2251)	5,6,7		(See Component 5.)	Required only if a problem is identified
Problem Report Log Form (F–2252)	5,6,7		(See Component 5.)	Required only if Problem Reports have been submitted
Rollout Report Form (F–5255)	5,6		(See Component 5.)	Required only if needed
Target Users Capability Upgrade Request Form (F–6054)	4,6		(See Component 4.)	Required only if needed
Customer Satisfaction Form (F–6055)	5,6		(See Component 5.)	Required
Final User Signoff Form (F–6056)	5,6		(See Component 5.)	Required

Product Name	Created in Component	Updated in Components	Product Summary	Required Product?
Go or No GO Decision for Project Form, Component 6 (F–1156)	6		Use this form to request approval to terminate maintenance support for an application system.	Required

Table C–7. Products Created or Updated within Component 7

Product Name	Created in Component	Updated in Components	Product Summary	Required Product?
Project Management Plan (S–1052)	1	2,3,4,5,6,7	(See Component 1.)	Required
Software Engineering Notebook (S–3091)	3	4,5,6,7	(See Component 3.)	Required
Records Retention and Disposition Authority (NRC Form 331)	1,7		(See Component 1.)	Required
Request for Records Disposition Authority (SF 115)	1,7		(See Component 1.)	Required
Enterprise Model Change Request Form (F–2070)	1,3,7		(See Component 1.)	Required
System Inventory Update Form (F–2071)	5,7		(See Component 5.)	Required
Technical Reference Model Update Form (F–2073)	3,5,7		(See Component 3.)	Required
Government Information Locator System (GILS) Update Form (F–3060)	5,7		(See Component 5.)	Required
Problem Report Form (F–2251)	5,6,7		(See Component 5.)	Required only if a problem is identified
Problem Report Log Form (F–2252)	5,6,7		(See Component 5.)	Required only if Problem Reports have been submitted
Go or No GO Decision for Project Form, Component 7 (F–1157)	7		Use this form to document that the system has been decommissioned.	Required

Appendix D. Activity Summary

D.1 Component Review

This appendix summarizes the activities of the seven components of NRC's SDLCM Methodology and specifies which activities are required. Use this appendix together with Appendix C, which summarizes the products of NRC Projects, as a cross-reference to the component descriptions in the main body of this handbook.

This appendix provides only *summary* checklists of the high-level activities described in the main body of this handbook. See Chapter 6 through Chapter 12 for details. Use the checklists to identify the activities that will be performed and the products that will be produced for a specific Project.

The SDLCM Methodology distinguishes among activities associated with (1) new development, (2) enhancement, and (3) maintenance of application systems. See Chapter 2, "Methodology Overview," and Chapter 11, "Component 6. Service the Solution," for more information. Figure D–1 shows the component structure introduced in Chapter 2 and relates that structure to the sections of this appendix.

- Section 2 summarizes the activities required for development of new application systems or enhancement of existing systems (Components 1–5).
- Section D.3 summarizes the activities performed for routine and emergency maintenance (Component 6).
- Section D.4 summarizes the activities required when decommissioning any operational system (Component 7).

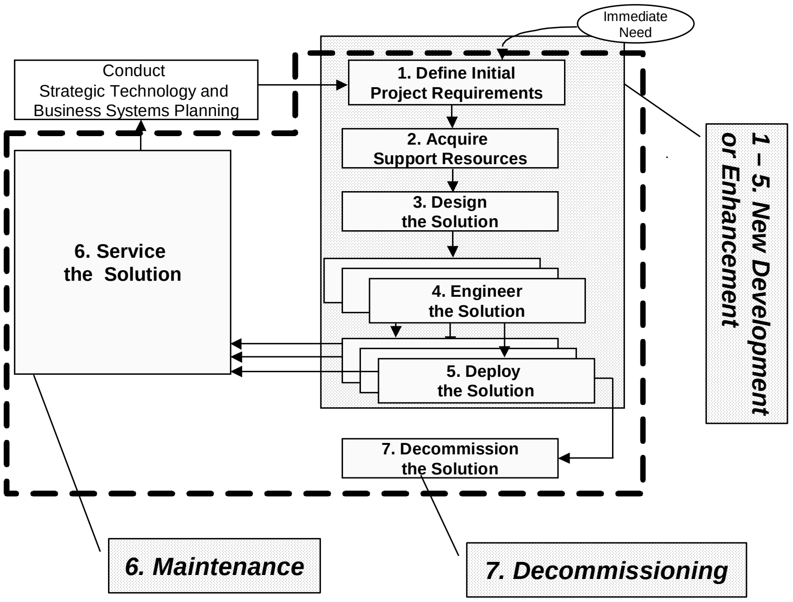

Immediate Need
Conduct Strategic Technology and Business Systems Planning
1. Define Initial Project Requirements
2. Acquire Support Resources
3. Design the Solution
4. Engineer the Solution
5. Deploy the Solution
6. Service the Solution
7. Decommission the Solution
1 – 5. New Development or Enhancement
6. Maintenance
7. Decommissioning

Figure D–1. Review of Component Structure

D.2 New Development and Enhancement

This section summarizes the activities performed for new development and enhancement of application systems. Chapters 6 through 10 provide the details. See Appendix C for a summary of the products that must be produced.

The five tables in this section can be used as checklists for tailoring the SDLCM Methodology to the specific needs of new development and enhancement Projects.

Remember: activities are not sequential steps. Although some activities cannot be started until a predecessor activity is complete, others can be performed concurrently. The diagrams in the main body of this handbook clarify the dependencies among activities.

Table D–1. Checklist for Component 1

	1. Define Initial Project Requirements (2 Pages)					
	Activities		Products		Activity Req'd	
	Act. ID	Activity	Products	Standards or Forms	New Develop-ment	Enhance-ments
□	1.1	Clearly identify information management problem	Project Charter	S–1051	X	
□	1.2	Clarify the Project scope	System Requirements Specification	S–3051	X	X
			• *Section 1.3 Scope*			
			System Operations Concept	S-3053	X	
□	1.3	Establish a Project Plan and Project Support Controls	Project Management Plan	S–1052	X	X
			Quality Assurance Plan	S-2001	X	X
			Configuration Management Plan	S-3501	X	X
			SDLCMM Deviation or Waiver Form	F-2010		
□	1.4	Establish the Software Development Plan	Software Development Plan	S-1057	X	X
□	1.5	Notify Records Management Branch to Review Records Management Requirements.	Government Forms	NRC Form 616	X	
				NRC Form 331	X	
				NA Form 14028	X	
				SF 115	X	
□	1.6	Identify requirements	System Requirements Specification	S–3051	X	X
			Supporting Standards			
			Data Models	S–3151	X	
			Process Models	S–3161	X	
			Context Diagrams	S–3162	X	
			Data Flow Diagrams	S–3163	X	
			Data Dictionary	S–3351	X	
			System Operations Concept	S–3053	X	
			Enterprise Model Change Request Form	F-2070	X	
□	1.7	Analyze alternatives	Alternatives Assessment Document	S-3054	X	X
			Current System Assessment Document	S-3052	X	
□	1.8	Review the toolkit	Software Development Plan	S-1057	X	X

	1. Define Initial Project Requirements (2 Pages)					
	Activities		**Products**		**Activity Req'd**	
	Act. ID	**Activity**	**Products**	**Standards or Forms**	**New Develop-ment**	**Enhance-ments**
□	1.9	Develop a support resource request.	Project Management Plan *Section 4 - Project Organization* *Section 5 - Work Breakdown Structure*	S–1052	X	X
			Software Development Plan	S-1057	X	X
			Support Resources Acquisition Request Form	F-1061		
			Environment Change Request Form	F-1601		
□	1.10	Plan for deployment.	Tactical Integration Plan *Sections for which information can be specified.*	S–5051	X	X
□	1.11	Review the Project to make a Go or No-Go decision	Go or No-Go Decision Form	F–1151	X	X

Table D–2. Checklist for Component 2

	2. Acquire Support resources					
	Activities		Products		Activity Req'd	
	Act. ID	Activity	Products	Standards or Forms	New Development	Enhancement
□	2.1	Specify the work to be done	Statement of Work	S–1053		
□	2.2	Staff the Project	Project Management Plan • Section 4 Project Organization	S–1052	X	X
□	2.3	Train the staff	Developer Training Requirements Form	F–7051		
			Fully Staffed and Trained Team Form	F–1181	X	X
□	2.4	Acquire and install other required resources	Development and Maintenance Environment Products Installation Plan Support Resource Acquisition Strategy Form	S–1055 F–1062		
□	2.5	Update the Project Management Plan	Project Management Plan • Section 5 Work Breakdown Structure • Section 1.4.2 Risk Management • Other sections that require changes	S–1052	X	X
□	2.6	Continue Deployment Planning	Tactical Integration Plan • Provide additional information and update any that has changed.	S–5051	X	X
□	2.7	Review the Project to make a Go or No-Go decision.	Go or No-Go Decision Form	F–1152	X	X

Table D–3. Checklist for Component 3

	3. Design the Solution (2 pages)					
	Activities		Products		Activity Req'd	
	Act. ID	Activity	Products	Standards or Forms	New Develop-ment	Enhance-ment
□	3.1	Analyze requirements and perform high level design	System Requirements Specification • Updates to any information that has changed.	S–3051	X	X
			System Operations Concept	S–3053	X	X
			Logical Design Document	S–3171	X	X
			Physical Design Document	S–3172	X	X
			Supporting Standards			
			Data Models	S–3151,	X	
			Process Models	S–3161,	X	
			Context Diagrams	S–3162,	X	
			Data Flow Diagrams	S–3163,	X	
			Data Dictionary	S–3351	X	
			Software Engineering Notebook	S–3091	X	X
			Enterprise Model Change Request Form	F–2070	X	X
□	3.2	Plan solution integration	Solution Integration Plan	S–5053	X	X
			or Tactical Integration Plan	S–5051		
□	3.3	Design training materials	Integrated Education, Training, and Reference Materials	S–7052	X	
□	3.4	Establish test approach	Test Plan	S–5151	X	X
□	3.5	Update the Project Management Plan and Software Development Plan	Project Management Plan	S–1052	X	X
			Software Development Plan • (Updates to any information that has changed.)	S-1057	X	X

	3. Design the Solution (2 pages)					
	Activities		Products		Activity Req'd	
	Act. ID	**Activity**	**Products**	**Standards or Forms**	**New Develop-ment**	**Enhance-ment**
□	3.6	Continue deployment planning	Tactical Integration Plan	S-5051	X	X
			Provide any additional deployment information and update any that has changed, either within the TIP or address as separate documents.			
			Solution Integration Plan	S–5053	X	X
			Conversion Plan	S–1054	X	X
			Security Plan	S–1056		
			Products Installation and Integration Plan	S–5052		
			Other Systems Integration Plan	S–5054	X	X
			Other Integration Issues Plan	S–5055		
			Technical Reference Model Change Request Form	F–2073	X	X
□	3.7	Review the Project to make a Go or No-Go decision.	Go or No-Go Decision Form	F–1153	X	X

Table D–4. Checklist for Component 4

	4. Engineer the Solution (2 pages)					
	Activities		Products		Activity Req'd	
	Act. ID	Activity	Products	Standards or Forms	New Develop-ment	Enhance-ment
□	4.1	Maintain the change log	Change Log Form	F–2561	X	
			Change Proposal Form	F–2502		
□	4.2	Engineer the detailed design	System Requirements Specification	S–3051	X	X
			Project Management Plan (update if necessary)	S-1052		
			Software Development Plan (update if necessary)	S-1057		
			System Operations Concept (update if necessary)	S-3053		
			Alternatives Analysis Document (update if necessary)	S-3054		
			Logical Design Document (update if necessary)	S–3171		
			Physical Design Document (update if necessary)	S–3172		
□	4.3	Build the solution	Software Engineering Notebook	S–3091	X	X
			Tactical Integration Plan or individual document:	S-5051		
			- Network Integration Diagrams (update if necessary)	S-5056		
			- Solution Integration Plan (update if necessary)	S-5053		
			- Other Systems Integration Plan (update if necessary)	S-5054		
			- Other Systems Issues Integration Plan (update if necessary)	S-5055		
□	4.5	Test the solution	Test Plan (Actual Results)	S–5151	X	X

	4. Engineer the Solution (2 pages)					
	Activities		**Products**		**Activity Req'd**	
	Act. ID	**Activity**	**Products**	**Standards or Forms**	**New Development**	**Enhancement**
□	4.6	Create the rollout strategy and continue deployment planning	Tactical Integration Plan (Section 2, Rollout Plan)	S–5051	X	X
			• Installation Instructions (or as appendix to TIP)	S–5252		
			• Network Integration Diagrams (in TIP)	S–5056		
			• Additional TIP information/update with changes.			
			Target Users Capability Upgrade Request Form	F–6054		
□	4.7	Update Policies and Procedures	System Inventory Update Form	F-2071	X	X
			Technical Reference Model Update Form	F-2073		
			Government Information Locator System Update Form	F-3060		
			Management Directives, Rule Changes, Procedures, etc.			
□	4.8	Build the user material and training materials	User Guide	S–6051	X	X
			Online Help System and Tutorials	S–6052		
			Operational Support Guide	S–6151		
			User Training and Orientation Plan	S–7053		
□	4.9	Deliver the Training	User Guide	S–6051	X	X
			Online Help System and Tutorials	S–6052		
			Operational Support Guide	S–6151		
			User Training and Orientation Plan	S–7053		
□	4.10	Update the Project Management Plan and Software Development Plan	Project Management Plan	S–1052	X	X
			Software Development Plan	S-1057		
□	4.11	Review the Project to make a Go or No-Go decision.	Go or No-Go Decision Form	F–1154	X	X

Table D–5. Checklist for Component 5

	5. Deploy the Solution (2 pages)					
	Activities		Products		Activity Req'd	
	Act. ID	**Activity**	**Products**	**Standards or Forms**	**New Develop-ment**	**Enhance-ment**
□	5.1	Review and update plans and announce deployment	Project Management Plan	S–1052	X	X
			Tactical Integration Plan	S–5051		
			·Updates to any information that has changed.			
			Deployment Announcement		X	X
□	5.2	Validate and upgrade the environment	(Current Environment)		X	X
			Tactical Integration Plan	S–5051		
□	5.3	Install the solution	Installation Instructions	S–5254	X	X
			Installation Report Form	F–5254	X	X
□	5.4	Test the installed solution	(Tested solution)		X	X
	5.6	Operational Support System Test				
	5.10	User Test the Solution				
			Tactical Integration Plan	S–5051		
			Test Plan *(w/Test Log)*	S-5151	X	X
□	5.5	Analyze the Installed System Test Results	Problem Report Form	F–2251	X	X
	5.7	Analyze Operational Support Test Results				
	5.11	Analyze the User Test Results				
			Problem Report Log Form	F–2252		
			Performance Report Form	F–5256		
			Deployment Recommendations for Fix Form	F–5257		
□	5.8	Implement policies and procedures	System Inventory Update Form	F–2071	X	
			Technical Reference Model Update Form	F–2073		
			GILS Update Form	F–3060		
			(Updated system-specific policies and procedures)			
□	5.9	Train the users	Change Proposal Form	F–2502		
			Problem Report Form	F–2251		
			Trained Users Form	F–7151		

	5. Deploy the Solution (2 pages)					
	Activities		Products		Activity Req’d	
	Act. ID	Activity	Products	Standards or Forms	New Development	Enhancement
□	5.12	Acceptance test the solution	Software Engineering Notebook	S–3091	X	X
			Customer Satisfaction Form	F–6055		
			Deployment Recommendation for Fix	F-5257		
			Final User Sign-off Form	F–6056		
			Change Proposal Form	F–2502		
			Problem Report Form	F–2251		
□	5.13	Analyze the Acceptance Test Results	Problem Report Form	F–2251	X	X
			Problem Report Log Form	F–2252		
			Solution Modules Ready for Deployment Form	F–4051		
			Deployment Recommendations for Fix Form	F–5257		
□	5.15	Deploy the Solution	Rollout Report	F-5255	X	X
			Cutover Announcement		X	X
□	5.16	Review Component 5.	Go or No-Go Decision Form	F–1155	X	X

D.3 Maintenance

This section summarizes the activities performed for routine and emergency maintenance. Chapter 11, “Component 6. Service the Solution,” provides the details.

Either a Change Proposal or a Problem Report initiates system maintenance. The Change Management activities are always required. Most changes will be handled as routine. Only true emergencies should be treated as described in Section 11.5.

The products are not specified, because they are strongly dependent on the nature of the maintenance change. See Appendix C for a product summary.

Table D–6. Checklist for Component 6

	6. Service the Solution				
	Activities		Products		Activity Req’d
		Activity	**Products**	**Standards or Forms**	**Maintenance**
	Change Management				X
	□	Establish change management			X
	□	Control system changes			X
	□	Track and report system changes			X
Select either routine or emergency maintenance					
□	Routine Maintenance				
	□	Release Management			
	□	Plan the release			
	□	Coordinate release changes			
	□	Track release changes			
	□	Track releases			
	□	Release-Based Maintenance			
□	Emergency Maintenance				
Select the following if the system will no longer be supported					
□	Terminate Maintenance				
	□	Review the Project to make a Go or No-Go decision.	Go or No-Go Decision Form	F–1156	X

D.4 Decommissioning

This section summarizes the activities required when decommissioning any operational system. See Chapter 12, “Component 7. Decommission the Solution,” for details.

Table D–7. Checklist for Component 7

	7. Decommission the Solution				
	Activities		**Products**		**Activity Req’d**
	Act. ID	**Activity**	**Products**	**Standards or Forms**	**All Systems**
☐	7.1	Review and update plans and announce decommissioning	Project Management Plan (Updates to any information that has changed.) Decommissioning Announcement	S–1052	X
☐	7.2	Notify Records Management Branch to review Records Management requirements	Records Retention and Disposition Authority	NRC Form 331	X
☐	7.3	Analyze system interfaces and document effect on any other systems (It is the responsibility of the owners of other systems to respond to the PRs if necessary.)	Problem Report Form Problem Report Log Form	F–2251 F–2252	X
☐	7.4	Obtain approval to remove the solution from the operational environment	Go or No-Go Decision Form (Business Advocate signature)	F–1157	X
☐	7.5	Update documentation and records	Request for Records Disposition Authority	SF 115	X
			Enterprise Model Change Request Form	F–2070	
			System Inventory Update Form	F–2071	
			Technical Reference Model Update Form	F–2073	
			GILS Update Form	F–3060	
			Software Engineering Notebook (Updated system-specific policies and procedures)	S–3091	
☐	7.6	Wait for confirmation that any affected systems have been re-deployed and remove the system from the operational environment	(Operational environment with any previous external interfaces removed)		X
☐	7.7	Obtain final sign-off that decommissioning is complete	Go or No-Go Decision Form (Executive Sponsor signature)	F–1157	X

Appendix E. Transition of Legacy Systems

This appendix summarizes the process for transition of legacy systems from the NRC to a contractor for life-cycle management.

E.1 Systems Developed with SDLCM Methodology Guidance

This section addresses the case in which NRC or a contractor has developed an application system under the guidance of NRC's SDLCM Methodology. Under these conditions, when NRC transitions the application system to a different NRC organization or contractor for maintenance or enhancement, NRC will provide the system source code plus printed and electronic copies of all available system documentation, including at least the following:

- System Requirements Specification
- System Operations Concept
- Analysis of Alternatives
- Current System Assessment Document
- Software Engineering Notebooks
- Logical Design Document (The following documents must be included or provided separately.)
  - Data Models
  - Process Models
  - Context Diagrams
  - Data Flow Diagrams
  - External Systems Interface Diagrams
  - Data Dictionary
- Physical Design Document (The following documents must be included or provided separately.)
  - Data Models
  - Process Models
  - Context Diagrams
  - Data Flow Diagrams
  - External Systems Interface Diagrams
  - Data Dictionary
- Tactical Integration Plan (The following documents must be included or provided separately.)
  - Operations and Maintenance Plan
  - Products Installation and Integration Plan

  - Solution Integration Plan
  - Other Systems Integration Plan
  - Other Integration Issues Plan
  - Network Integration Diagrams
- Test Plan
- Integrated Education, Training, and Reference Materials
- Operational Support Guide
- User Guide
- On-Line Help System and Tutorials
- Waivers

The organization responsible for supporting the life-cycle management of the application system will maintain the system documentation to be consistent with changes in the operational system.

E.2 Systems Developed without SDLCM Methodology Guidance

This section addresses the case in which 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. This situation may arise if the NRC wishes to transition a legacy system developed prior to NRC's introduction of the SDLCM Methodology or if a user who is unaware of the requirement to adhere to NRC's mandated system development methodology develops a new system.

Under those conditions, when NRC transitions the application system to a different NRC organization or contractor for maintenance or enhancement, NRC will provide the system source code plus printed and electronic copies of all available system documentation, including those listed if the preceding section, if possible.

A minimal subset of the system documentation is critical for the successful maintenance of any application. Without this minimal subset of system documentation, there is a high risk that the maintenance organization will be unable to satisfy requests for changes to the application system. If that subset is not available, NRC will fund the maintenance organization to develop the following minimal documentation at the time of transition:

- System Requirements Specification
- Software Engineering Notebook(s)
- Logical Design Document (portions):
  - Data Flow Diagrams
  - Data Dictionary
- Physical Design Document (portions):
  - Data Flow Diagrams
  - Data Dictionary

- Tactical Integration Plan (portions or optional subdocuments):
  - Solution Integration Plan
  - Other Systems Integration Plan
  - Network Integration Diagrams
- User Guide

Table E–1 summarizes the system documentation required for transition of an application system and references all of the products listed in Section E.1. For the convenience of the reader, the table also indicates the identifier of the SDLCM Methodology standard that defines the content of the each product. For potentially complex documents (that is, the SRS, LDD, PDD, and TIP), the table lists sections (or optional subdocuments in the case of the TIP) separately.

The table also indicates the risk that maintenance will not be successful if the documentation is not provided.

- **High**. The documentation is *required* for successful maintenance. NRC must minimally provide the indicated products at the time of transition or must fund the maintenance organization to develop the products at that time.
- **Medium**. The documentation is *desirable*. If it is not provided, NRC will fund development of the documentation *when* is it required for specific maintenance or enhancement activities.
- **Low**. The documentation is *potentially useful* and will be provided to the maintenance organization if it is available. Although this documentation is essential when developing a new system, it is unlikely to be essential during maintenance. NRC will fund the creation of the documentation *if* it is needed for successful maintenance.

E.3 Alternative System Documentation

NRC's SDLCM Methodology provides an alternative standard to be used only for the case in which 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. In such a situation, preparing the complete SRS, LDD, and PDD after the application system has already been developed may not be useful. Portions of those products, although needed during the initial development of the application, may not be essential for system life-cycle management.

SDLCM Methodology Standard S–4151, 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. (See the standard for details of the content.) As the application system is modified through maintenance or enhancement, the maintenance organization will also maintain the as-built documentation to reflect the modified requirements and design.

A decision by NRC to adopt the As-Built System Documentation for transition of a legacy system that was not developed with SDLCM Methodology guidance simplifies the minimal subset of the system documentation that is critical for the successful maintenance of an application. Instead of the list prescribed in Section E.2, NRC may elect to fund the maintenance organization to develop the following minimal documentation at the time of transition:

- As-Built System Documentation
- Software Engineering Notebook(s)
- Tactical Integration Plan (portions or optional subdocuments):
  - Solution Integration Plan
  - Other Systems Integration Plan
  - Network Integration Diagrams
- User Guide

Table E–1. Documentation Required for System Transition

Project Product Name (specified by NRC's SDLCM Methodology)	Identifier	Risk if not Provided		
		High	Med	Low
System Requirements Specification	S–3051	✓		
Current System Assessment Document	S–3052		✓	
Analysis of Alternatives Document	S–3054			✓
System Operations Concept (SOC)	S–3053		✓	
Software Engineering Notebook	S–3091	✓		
Logical Design Document	S–3171	✓		
Data Models (Logical)	S–3151		✓	
Process Models (Logical)	S–3161		✓	
• Context Diagram (Logical)	S–3162		✓	
• Data Flow Diagrams (Logical)	S–3163	✓		
• External Systems Interface Diagrams (Logical)	S–3164		✓	
• Data Dictionary (Logical)	S–3351	✓		
Physical Design Document	S–3172	✓		
• Data Models (Physical)	S–3151		✓	
• Process Models (Physical)	S–3161		✓	
• Context Diagram (Physical)	S–3162		✓	
• Data Flow Diagrams (Physical)	S–3163	✓		
• External Systems Interface Diagrams (Physical)	S–3164		✓	
• Data Dictionary (Physical)	S–3351	✓		
Tactical Integration Plan (and related sub-documents)	S–5051			
• (Section 2) Rollout Plan	S–5051			✓
• (Section 3) Operations and Maintenance Plan	S–5051		✓	
• Conversion Plan	S–1054			✓
• Security Controls	S–1056			✓
• Products Installation and Integration Plan	S–5052		✓	

Project Product Name (specified by NRC's SDLCM Methodology)	Identifier	Risk if not Provided		
		High	Med	Low
• Solution Integration Plan	S–5053	✓		
• Other Systems Integration Plan	S–5054	✓		
• Other Integration Issues Plan	S–5055		✓	
• Network Integration Diagrams	S–5056	✓		
• Installation Instructions	S–5252			✓
Test Plan	S–5151		✓	
Integrated Education, Training, and Reference Materials	S–7052			✓
Operational Support Guide	S–6151		✓	
User Guide	S–6051	✓		
On-Line Help System and Tutorials	S–6052		✓	

Appendix F. Project Management Overview of SDLCMM

This appendix is written to present the SDLCMM from the view point of the Overall Project Manager's responsibilities. It provides scaling differences that exist between Maintenance Projects and Development/Enhancement Projects. It also provides information regarding SDLCMM documentation developed during the CPIC Selection Phase that is necessary for the CPIC Business Case development and approval to move forward to the CPIC Project Phase.

F.1 The Overall Project Manager and the Use of the SDLCMM

The Overall Project Manager and his Project Team are responsible for the success of the Project. The Systems Development Life-Cycle Management Methodology (SDLCMM) is established as a Handbook which provided guidance regarding controls, decisions, and activities for a generic project. It is supported by a volume containing Standards, Procedures and Forms.

The SDLCMM Handbooks defines seven Components. While the general life-cycle of any Project resulting in a system which includes automation of some or all of the processes, begins with Definition (Chapter 6 - Component 1) and ends with Decommissioning (Chapter 12 - Component 7), the greater period of time for use of a system is spent in Service/Maintenance (Chapter 11 - Component 6). It is important to note that the reason for defining the SDLCMM in "components" rather than "phases" is because it is more flexible in this format to applying it to the various Software Development Life-Cycle Models (See Chapter 3 of the SDLCMM Handbook). The Overall Project Manager, is responsible for choosing early in the Project the Software Development Life-Cycle Model that will be used for the Project. He makes this decision with input from his Project Team, specifically, the Technical Project Manager, based on the detail level of understanding known of the requirements, the funding approach available for the project, etc.

The Overall Project Manager is responsible for the planning, management, and oversight of the Project. He does this through effective use of Project Management skills and appropriate delegations to his Project Team members. He must ensure that the staffing is adequate for the various roles necessary to support the Project. A definition of the roles and the responsibilities of these roles can be found in Appendix B of the SDLCMM Handbook. While at the beginning of the Project the roles of Executive Sponsor and Overall Project Manager may be the only staffed positions, it is important to note that the SDLCMM provides for planning for and acquiring (Chapter 7 - Component 2) the necessary skills to meet the other roles and responsibilities. The Overall Project Manager should ensure that all resources needed are addressed not just those that achieved through procurement or contracting. Go/No Go decisions are required even in Component 2. This means that all commitments for proper staffing should be completed before going forward with the Project.

The Project's requirements will be refined from the initial requirements and systems operational concept and will reflect a lower level of detail at the time of design from that known at the beginning of the Project. For this reason, it is important the Overall Project Manager address change management and version control early in the Project so that proper controls are in place as changes are made. Information on Configuration Management (which includes Change

Management and Version Control of products of the Project) is contained in Chapter 5 of the SDLCMM Handbook and the procedure for using a Change Control Board approach is defined in the Companion Volume as P-2501.

Additionally, the quality of the Project's products will need to examined throughout the Project to ensure a high quality product at the end and traceability back to the original requirements. Information on Quality Assurance is provided in the SLDCMM Handbook in Chapter 4.

A list of the required products and the associated activities is provided in a quick checklist form in Appendix D of the SDLCMM Handbook. This should be reviewed and understood by the Overall Project Manager and any plans to omit specific products or substitute others should be addressed early on through the submission of a formal Request for Deviation or Waiver from the SDLCMM. The form that is used to document this decision and obtain approval by the Office of the Chief Information Officer is F-2010 which can be found in the SDLCMM Companion Volume of Procedures, Standards and Forms.

The OCIO provides formal training on the SDLCMM and is available to assist Overall Project Managers with any questions regarding the Methodology. It is a requirement that Overall Project Mangers be trained in the SDLCMM and is recommended that all members of the Project Team receive training.

In sections that follow there is a discussion of Components 1 through 5 which may be useful from a Overall Project Manager perspective when managing either a development or maintenance project/effort. It presents in a narrative format the same information provided in the Handbook Chapters 6 - 10, but provides some additional information related to use of the Methodology for Maintenance activities and the relationship of the activities and products to the CPIC process.

F.2Component Discussions

F.2.1 Component 1

During Component 1, system (or subsystem) concepts are defined and selected, and the technical groundwork is laid for the entire Project. This phase may be:

- The initial phase of a development effort
- The beginning of an effort which includes some or all of the following: definition, analysis, design, development/engineering and deployment/delivery of a system, subsystem(s) or components for new installations, or as modification/improvement to an existing system
- A study effort which completes when the requirements are analyzed and results are reported

In any of these cases, SDLCMM Component 1 activities involve reviewing and evaluating Project system requirements and translating them into alternative operating concepts and specifications. All requirements, at all levels, must be reviewed by all appropriate members of the Project Team, especially the Overall Project Manager, Technical Project Manager and Business Advocates, before being incorporated into the Project. Functional analysis, identifying interfaces, synthesis, and analysis/tradeoff studies are the primary Component 1 activities which provide the basis for allocating requirements, developing alternative design concepts, and for refining the System Requirements Specification (SRS) in follow-on Components. The SRS contains the baselined requirements for the system. Component 1 effort completes with a System Requirements Review (SRR). The system configuration, including requirements, that is established at the SRR represents the Functional Baseline. The procedure and description of the SRR is contained in P-2010. Project Team participation and approval at all appropriate levels and for all disciplines is required in the performance of the SRR.

Formal documentation produced in this component includes: Project Charter (S-1051), preliminary Project Management Plan (S-1052), Current System Assessment Document (S-3052), System Operations Concept (S-3053), System Requirements Specification (S-3051), Configuration Management Plan (S-3501), Quality Assurance Plan (S-2001), Software Development Plan (S-1057), and Analysis of Alternatives (S-3054) for approaching the development of a solution and identification of support resources needed to proceed with the Project.

Component 1 activities may start as a part of CPIC Selection Phase activities related to responding to requests for new products and major product enhancements. It may, alternatively, start with a change to an existing system in the form of an adaptive, corrective, preventive maintenance or a perfective change viewed as a minor enhancement. The specifics of the need, or need for change, must be communicated clearly to all concerned parties and to NRC management in order to make timely and informed decisions. The early efforts called for in the CPIC Selection Phase, including planning (Components 1, 2 and 3), analysis (Components 1 and 3), review (Components 1 and 3), and design (Component 3), provide the basis for an accurate cost estimate and improve the chances for performing to specifications on time and within budget.

Projects that require CPIC approval are, typically, development Projects (i.e., new work resulting in a new application system) or enhancement to an existing application system, as opposed to maintenance efforts where a change, or minor enhancement, is being made to an existing system. This section will address the Component 1 process to be performed under each starting scenario but will tie together coordinating activities from Components 2 and 3. The specific Component 2 and 3 activities will be addressed in more detail in follow-on sections.

F.2.1.1 Starting a Project under CPIC

While the CPIC Selection Phase starts with Component 1, where requirements are defined and analyzed, it does not end until a Business Case is defined. The definition of the Business Case requires that activities defined in Components 2 and 3 also be accomplished. Products that will be completed in Components 2 and 3, which need to be initially addressed on Component 1 at a high level, are discussed as well.

In the CPIC Selection Phase, Component 1 defines the requirements by breaking down the statement of need (information from the CPIC Screening Form) into functional, performance, operational, programmatic, special (accessibility, records, etc) and data requirements; in the Component 1 activities a definition of the requirements is developed and a concept of operations is produced which provides a description of the system. The concept (Systems

Operations Concept) is used as a tool to verify that the requirements list (System Requirements Specification) is complete and to provide input regarding details of the programmatic use and processes from a user perspective when performing Component 3 activities. An additional Component 1 activity completed during this CPIC Phase involves performing an analysis of alternatives for approaching the development of the system that will satisfy the concept and meets the requirements. This analysis provides initial input to the development of the Business Case in the form of initial cost/benefit figures and analysis of the return on investment. The alternatives analysis generally is conducted on at least three alternatives: staying as is (current system assessment forms this baseline), and two or more other alternatives including such approaches as: modifying an existing system, building a new system from the ground up, or using COTS, etc. Detailed Project Planning is started during the Component 1 effort and is refined as the requirements are baselined and the Project progresses in Components 2 and 3. Baselining requirements at the end of the Component 1 effort is key to establishment of the functional capabilities of the proposed system.

The portions of the CPIC Selection Phase begun during Component 1 are:

- Requirements Identification and Definition
- Alternatives Analysis with Cost/Benefit/Risk Analysis
- Development of the Project Management Plan
- Development of a System Operations Concept document

The early Systems Requirements Identification and Definition, Alternatives Analysis, and Project Planning effort is considered a vital stage of the SDLCMM process for hardware/software development or system integration Projects because it is here that the entire Project Team (made up of NRC staff and others, as required, including the Executive Sponsor, Business Advocates and Business Subject Matter Experts) reaches agreement on foundational elements of the Project prior to actual design efforts. It is a key tenet of the SDLCMM that the personnel expected to perform the primary roles in the development and acceptance of the application system are part of the planning effort.

The purpose of the early Systems Requirements Identification and Definition, Alternatives Analysis, and Project Planning effort is threefold: (1) to allow the Executive Sponsor and the Business Advocates to communicate functional, performance, operational, programmatic, and other specific requirements from their own point of view to each other to reconcile terminology and any conflicting objectives; (2) to document these requirements with definitions and examples to ensure that they are uniformly understood by the Project Team; and (3) to formulate a detailed plan for accomplishment of the required effort. These requirements expand on the CPIC Screening Form inputs and are captured in the initial draft of the SRS, while the planning and scheduling efforts are captured in the PMP and SDP.

Planning efforts set the stage for effective tracking, management and oversight of both the requirements and the actual planned Project effort. During planning, the organization, facilitation, work breakdown structure, program schedule, staffing needs, cost, and detailed plans necessary for the Project Team to meet the Project, or Program obligations in a cost-effective manner are established. It is imperative that planning be accomplished to ensure that adequate resources and funding are allocated in order to achieve cost effectiveness. During the Needs Analysis and Planning effort, the bringing together of all Project Team disciplines into a single team causes integration of efforts to produce a quality product as each follow-on components of the SDLCMM is executed.

> Note: It is implicit that the Overall Project Manager ensure that adequate resources and funding are obtained during the Component 2 effort to accomplish all necessary activities in accordance with NRC policy.

The goal of the System Requirements Definition and Alternatives Analysis effort is to formally document and obtain approval for the system requirements, as expressed and documented in the SRS; definition and agreement upon a concept of how the system will work; and the approach that will be taken to bring this system to fruition. The functional requirements for the system may be gathered through surveys, statements, feedback on prototypes, problem reports or interviews, etc. and are expanded into a complete composite set of system requirements (SRS). These system requirements reflect what is needed to support and deliver to the level of the system operational concept (SOC) and should be stated in terms of system capabilities, including performance; interfaces with other systems; physical characteristics; user business knowledge and computer skills; system quality factors, such as reliability, maintainability, and availability; flexibility and expansion; documentation; logistics; data; and personnel and training. The system analysis also determines the qualification requirements; that is, the qualification tests that are necessary to verify and validate the system capabilities. The Overall Project Manager is responsible, during this Component, to develop the User Acceptance test criteria which addresses those requirements that must be satisfied for user acceptance of the application system.

System Requirements Definition and Alternatives Analysis effort is the first step in the process of converting the requirements derived from the initial CPIC Screening Form (Statement of Need) into a combination of system elements which satisfy that need. It involves analysis of existing documentation and conducting discussions with the Project Team (Technical Interchange Meetings (TIMs)) to document agreement on requirements and to refine the purpose and the manner in which the user will operate the system. External interfaces to the system are identified, as are system capacities. Methods for requirements verification of each requirement statement are also determined at this point. The process iterates with functional analysis and synthesis efforts to assess feasibility and to structure to the requirements in the most cost effective manner. This iterative process serves to make the verification process more accurate and cost effective by eliminating ambiguity in the requirement statements. The Project Team should also strive to eliminate unnecessary implied design in the requirement statements.

A very large system may require a multi-tiered approach to the System Requirements Definition and Alternatives Analysis resulting in a system level requirements analysis and multiple subsystem level analyses. For example, a system may be divided into two or more subsystems with a requirements analysis being performed at the system and subsystem levels. In this case, the System Requirements Definition and Alternatives Analysis effort will be performed at a level of abstraction appropriate for a High-Level System Design (see Component 3). The High-Level System Design will identify the subsystems and allocate system requirements to each. The process is then repeated for each subsystem. The input to the System Requirements Definition and Alternatives Analysis effort for each subsystem then includes high-level system requirements produced by the system level requirements analysis effort. All requirements of the High-Level System Design must map to the appropriate subsystem design and to the appropriate identifier for management control within that subsystem. Traceability of all system requirements must be documented.

F.2.1.2 Starting Component 1 for a System under Maintenance

Maintenance efforts are changes, minor enhancements and fixes to existing systems (applications and their support environments). Maintenance is a Component 6 (Service the Solution) effort and is explained more fully in Section 11. This section will detail the Component 1 related activities associated with maintenance.

A maintenance effort is comprised of a reiteration of the necessary actions from Component 6 and Components 1 through 5, but the efforts are separate, i.e., not aggregated as in the CPIC process where Component 2 and 3 efforts are prerequisites to moving forward. In this scenario, Component 2 and Component 3 are separate efforts.

Maintenance is based upon a documented Change Management process. The Change Management process may be different for different applications/systems, but must be documented in order to effect consistency and repeatability. All Component 1 inputs must derive from the Change Management process. Inputs generally take the form of a Change Proposal (CP) or Problem Report (PR). PRs and CPs report *what* needs to be fixed and may report how the fix should be effected. In any case, the input of the high level need, or definition of the problem, provides the basis for Component 1 Systems Requirements Definition and Alternatives Analysis activities. In the case where the CP/PR contain information regarding *how* the change should be made, some, or all, of the Alternatives Analysis effort will have been completed.

Maintenance efforts present the greatest risk of error because the application system is in place and all changes to it need to be documented and managed so that the application system is not compromised as a result of the new work. Analysis of the need for change, the definition, coordination and management of the system requirements, and the careful planning of design, test and deployment activities is paramount. Complete testing to ensure that the change has only altered the functionality that was intended (regression testing) is of extreme importance in this regard, specifically when new or changed functionality is introduced. Just as in Development efforts (Starting a Project under CPIC), Component 1 activities during a Maintenance effort set the stage for all of the follow-on efforts by establishing a firm baseline which describes what is to be reflected in the changed product.

SDLCMM Maintenance scenarios occur under one of three circumstances:

- Release-Based Maintenance
- Minor Enhancement
- Emergency Maintenance

Release-Based Maintenance is the most common instance of a Component 6 Maintenance effort. It assumes that periodic, or aperiodic, planned releases will be performed on a given system through a Release-Based Management process. CPs and/or PRs are usually aggregated to form the planned deployment, but a release may be comprised of only one CP/PR. When properly executed, this option presents the lowest level of risk because there is, or should be, the highest level of Systems Requirements Definition and Alternatives Analysis of the current system performed. There will also be highest level of planning for making alternations to the application system, testing it, and coordinating it with the implementation efforts. With the communication of the Project Management Plan (PMP) and Software Development Plan (SDP), if applicable, all concerned or affected entities may be coordinated for requirements definition/management, engineering effort monitoring, completion and approval of all appropriate testing (including appropriate regression testing), training development and delivery and deployment of the updated system.

Minor Enhancement is a "mini" development effort, requires careful definition and management of requirements to ensure that the product baseline is protected, and presents the greatest risk of missing a step in the SDLCMM process. It does not however, when properly executed, present the greatest risk to the delivered product, simply because it is well planned. This case requires the most effort because it will transition through all aspects of the SDLCMM define, design, engineer and deploy activities. It is, essentially, a repeat of the process that a CPIC Project would use, without the up-front emphasis on building the business case. Care must be taken, at minimum, to consider change to all documentation and to make a case-by-case judgement as to whether each document needs to be changed/updated. Some documentation is required, such as the PMP and TIP, updates to the SRS, and, in later components, Logical Design Document and Physical Design Document. Planning for the update of all appropriate documentation is a must.

Emergency Maintenance presents the highest risk to the system product due to the need for immediacy of change. Compromises may be made in the heat of getting a system back on-line after experiencing a fault which requires prompt action. In such a situation, it is prudent to make only absolutely necessary changes and to place them in a "temporary" location so that they may be easily removed after more permanent, properly engineered modifications are released through the release-based maintenance approach, explained above. Sometimes emergency fixes are effected through "patches" to the code to simplify the process. It is, however, required that the Component 1 Systems Requirements Definition and Alternatives Analysis activities be performed as soon as possible after system repair. Because actions are usually taken in haste, sometimes without regard to "correctness," it is crucial that the Maintenance team carefully document exactly what was done to address the emergency so that it is easier to assess the Systems Requirements Specification (SRS) and other documentation changes that will need to be made. It is important that the correct paperwork, i.e., Problem Report (PR), is generated to document the problem, that the PR be reviewed and approved at all appropriate levels, and that all appropriate documents, especially the System Requirements Specification, are updated as soon as possible to reflect the modified state of the system. The SRS must be updated as part of the Component 1 Maintenance effort as the functional baseline has changed. There may also be a need for regression testing to ensure that the "fix" will not, actually or potentially, cause other problems with the changed, or an interfacing, system.

Component 1 for any scenario in Maintenance includes the Systems Requirements Identification and Definition, Alternatives Analysis and Project/Effort Planning aspects of the SDLCMM process. It focuses on discerning the actual change required, planning the effort to accomplish the changes, and updating all appropriate documentation to ensure that adequate knowledge is transferred to continue maintenance of the system. The most important outcome of Component 1 effort is the maintenance of the functional baseline as reflected in the requirements contained in the SRS. It is, therefore, of extreme importance that attention be given to updating the SRS as early as possible in the Component 1 Maintenance effort, no matter which scenario.

Component 1 defines the requirements by breaking down the statement of need (information from the Change Proposal (CP)/Problem Report (PR), i.e., new reporting format requirements, description of a problem experienced, etc.) into changes to the functional, performance, operational, programmatic, special (accessibility, records, etc.) and data requirements, which then become the updates for the System Requirements Specification. Component 1 then performs an analysis of the requirements and produces updates/changes to the System Operations Concept. Re-Baselining requirements at the end of the Component 1 effort is key to maintenance of the functional baseline (full list of satisfied and new requirements to be met by the application system).

The Systems Requirements Identification and Definition, Alternatives Analysis and Project/Effort Planning efforts are considered vital in the SDLCMM process for all types of maintenance efforts because it is here that the entire Maintenance Team (CLIN Manager, User/Sponsor, and Technical Subject Matter Experts) begins its planning and definition and analysis of the proposed system changes prior to commencement of the design efforts to effect them. If the Change Proposal (CP)/Problem Report (PR) that is driving the effort has already performed the analysis and/or defined the design solution, then that analysis/definition must be communicated and approved by all appropriate parties (CLIN Manager, CCB Chair, etc.) before proceeding. It is also a key tenet of the SDLCMM that the maintenance personnel expected to perform the task participate in the planning activities of the Component.

The purpose of the Systems Requirements Identification and Definition, Alternatives Analysis and Project/Effort Planning efforts is threefold: (1) to allow the Executive Sponsor and the Business Advocates to communicate changes to functional requirements from their own point of view to each other to reconcile terminology and any conflicting objectives; (2) to document these changed requirements with definitions and examples to ensure that they are uniformly understood by the Project Team; and (3) to formulate a detailed plan for accomplishment of the required effort. The functional requirements discerned and/or expressed at this time may expand on the Change Proposal (CP)/Problem Report (PR) inputs and are captured in the update of the System Requirements Specification (SRS), while the planning and scheduling activities are captured in the Project Management Plan (PMP) for the maintenance effort. It is very important to not let the updating of the SRS be neglected in any of the Maintenance scenarios. In the case of Emergency Maintenance, the “after the fact” conduct of the Component 1 effort, i.e., after the actual fix has been made (even though only temporary), may lessen concern for ensuring that the new functional baseline is properly documented.

Planning activities set the stage for effective tracking, management and oversight of the requirements and the actual planned maintenance effort. During planning, the organization, facilitation, work breakdown structure, maintenance/enhancement schedule, staffing needs, cost, and detailed plans necessary for the Maintenance Effort Team to meet the maintenance effort obligations in a cost-effective manner are established. It is imperative that planning be accomplished to ensure that adequate resources and funding are allocated in order to achieve cost effectiveness. During the Systems Requirements Identification and Definition, Alternatives

Analysis and Project/Effort Planning activities, the bringing together of all Project Team disciplines into a single team causes integration of efforts to produce a quality product as each follow-on component of the SDLCMM is executed.

> Note: It is implicit that the Overall Project Manager (in this case, the individual also serving as Contract Office Technical Representative (COTR) or CLIN Manager) ensure that adequate resources and funding are approved during the Component 2 effort to accomplish all necessary activities defined in Components 2, 3, 4, 5 and 6 in accordance with NRC policy.

The purpose of the System Requirements Definition and Alternatives Analysis effort is to formally document and obtain approval for the changed system requirements, as expressed and documented in the Systems Requirements Specification (SRS) update; and the approach that will be taken to bring this maintenance effort to fruition. The changed system requirements include updates to the system operational concept (SOC) and should be stated in terms of system capabilities, including performance; interfaces with other systems; physical characteristics; system quality factors, such as reliability, maintainability, and availability; flexibility and expansion; documentation; logistics; data; and personnel and training. There is a need here for requirements to be prioritized and that criteria be defined for qualification tests that are necessary to verify and validate the changed system capabilities, as well as regression testing, to ensure that unchanged capabilities are intact.

System Requirements Definition and Alternatives Analysis is the first step in the process of converting the requirements derived from the initial Change Proposal (CP)/Problem Report (PR) into a combination of system elements which satisfy that need. It will involve analysis of existing documentation and may include conducting discussions with the Project Team (Technical Interchange Meetings (TIMs)) to document agreement on changed/new requirements and to refine the purpose and the manner in which the end-user will utilize the system. External interface changes are identified, as are system capacity changes, as applicable. Methods for verification and test of each changed/new requirement statement are also determined at this point. The process may iterate with functional analysis and synthesis efforts to ensure success of the maintenance effort through the verification process and by eliminating ambiguity. The Maintenance Effort Team should also strive to eliminate unnecessary implied design.

F.2.2 Component 2

The purpose of the Acquire Support Resources Component is to take action to develop and execute appropriate acquisition methods to secure the identified resources from Component 1, 3, 4, 5, 6 and 7. While activities within Component 2 may begin concurrent with activities within Component 1, these activities may not finish until after the Systems Requirements Review and Go/No Go Decision are completed for Component 1. Additionally, since the Methodology is Component based, rather than phase based, this Component may be revised each time Component 1, or other Components, identify additional resources (personnel, equipment, software, etc.).

Formal documentation produced in this Component includes updates to the Project Management Plan (S-1052), Tactical Implementation Plan (S-5051), Software Development Plan (S-1057). Formal documentation developed during this Component may include: Development and Maintenance Environment Products Installation Plan (S-1055) and one or more Statement of Work (S-1053). Statements of Work may address the acquisition of such items as: training of development staff in new tools; training for users prior to testing and/or implementation of the application systems; hardware and software for the development environment; hardware and software for the operational environment; editors, writers, etc. needed to document policy and procedures; contractor services for the design phase of the application development; contractor services for the engineering and deployment phase of the application development; contractor services for the operational support phase of the application life cycle.

Component 2 may start as being part of the CPIC Selection Phase activities related to responding to requests for new products and major product enhancements. It may, alternatively, start with an adaptive, corrective, or perfective (minor enhancement) change under a maintenance effort.

F.2.3 Component 3

The purpose of the Design the Solution Component (Component 3) is to allocate the baselined system requirements to manual or automated processes and, if automated, to the hardware and software approaches envisioned as the solution set for the automated system. The system design decisions (Logical Design Document and Physical Design Document) made in this component have a major affect on subsequent components. For software development projects, this component identifies and defines the process paths through the system that input (stimuli or transactions) to the system would take to provide an output (responses or transaction results). These process paths are documented as Process Models. Through the mapping of requirements to defined processes and the identification of creation, receipt, storage, delivery and reporting points for data, further documentation is produced which defines Data (Data Models and Data Dictionary) and captures business rules. The initial planning activities for testing, training, integration and deployment are begun in Component 3 and result in a number of documents being started in this component (Tactical Integration Plan, Test Plans and scenarios, Solution Integration Plan, Integrated Training and Education and Reference Materials, etc.). Initial decisions regarding the appropriate levels of testing and when and who should participate in these tests are made during this component.

Formal documentation reviewed and updated in this component may, as necessary, include updates to:

- Project Management Plan (S-1052)
- Alternative Analysis Document (S-3054)
- System Requirements Specification (S-3051)
- System Operations Concept (S-3053)
- Software Development Plan (S-1057)

Formal documentation requirements for Component 3 includes a review of:

- Configuration Management Plan (S-3501)

- Quality Assurance Plan (S-2001)
- Current System Assessment Document (S-3052)

Formal documentation requirements for Component 3 also includes drafting of the Test Plan (S-5151), planning for the development of the Training and Operational Support products, as well as completing the baseline of both the Logical Design Document (S-3171) and Physical Design Document (S-3172).

> Note: Baselining the Logical Design Document and Physical Design Document includes generation or update and baselining of:
>
> - Data Models (S-3151)
> - Process Models (S-3161)
> - Context Diagrams (S-3162)
> - Data Flow Diagrams (S-3163)
> - Data Dictionary (S-3351)

Changes identified that affect the System Requirements Specification (S-3051), at this point, will be handled through the Change Control process since that document has been previously baselined and now resides in the Project-Level Configuration Management Library. The design leads to a better assessment of cost, schedule, and risk, which input to the CPIC business case. The CPIC business case is submitted at the end of this Component. With the completion of this Component and submission of the CPIC business case, the Project must now halt to await approval to enter the CPIC Project Phase.

F.2.3.1 Component 3 under CPIC

Component 3 activities are a part of the CPIC Selection Phase activities related to responding to requests for new products and major product enhancements. Alternatively, Component 3 activities are also a part of non-CPIC level enhancement and maintenance activities which start with a change to an existing system in the form of an adaptive, corrective, preventive or minor enhancement change.

If this Component 3 effort is part of a CPIC Selection Phase, the activities may initially be focused on defining the design to the point that a business case with sufficient confidence in the definition of the cost and schedule can be presented for approval. After Project Phase approval, a verification of the design is accomplished in Component 4 to ensure that the design is still solid and that there have been no changes in requirements or other perturbations that affect the final design. The start of planning for integration and deployment, and generation of the TIP, are also accomplished in this Component. The design becomes baselined at the System Design Review (SDR) as all parties review and approve it to move to development in the Engineer the Solution effort, Component 4. Baseline of the design and completion of the SDR also are essential to begin the CPIC Project Phase.

In a development effort, once the transition is made to Component 4, prototyping, computer modeling and/or testing may be undertaken to determine if the best design concept was actually chosen in the Component 3 design effort. This ensures that the critical elements can meet operational and support requirements, and, if indicated, allows return to Component 3 for refinement of the design concept.

F.2.3.2 Component 3 under Maintenance

If Component 3 is part of a Maintenance effort, activities will either focus on validating a solution that may have been supplied with the Change Proposal/Problem Report (CP/PR) that started the maintenance effort or on designing a solution to satisfy the need expressed in the corresponding CP/PR. The design becomes baselined at the System Design Review as all parties review and approve it to move to development in the Engineer the Solution component, Component 4.

F.2.3.3 Component 3 Either CPIC or Maintenance

This Component utilizes documentation products developed in other components as foundational documents to perform the actual design of a system to the point of approval for development. The initial objective of the component is validating the system functional baseline by establishing that it is complete and consistent. The second objective is to develop candidate architectures (designs, hardware platforms and vendor tools (COTS), selecting those which can fulfill the requirements within cost and schedule constraints.

> Note: In very large systems where a multi-tiered approach has been taken to System Requirements Analysis, resulting in a (High-Level) system requirements analysis and multiple (Lower-Level) subsystem analyses, the design process will be iterated for each subsystem. Extra care must be taken in these cases to ensure integrity of the system. Thus, interface requirements and design take on more importance. All requirements of the High-Level System Design must map to the appropriate subsystem design and to the appropriate lower level code or hardware solution within that subsystem. Traceability of all system requirements must be documented.

Conceptual design efforts and studies and tradeoffs of the alternatives are continued and/or refined until the requirements are synthesized into firm system design concepts. The resulting planned discrete pieces of the system (Configuration Items (CIs)) are documented in the Logical Design Document (LDD), Physical Design Document (PDD), Process Models (PM), Data Models (DM), Context Diagrams (CD), Data Flow Diagrams (DFD) and Data Dictionary (DD). Training and Operational Support materials are also to be included in the items developed and/or updated in this Component. The Software Development Plan (SDP) and other required software documentation are delivered, or updated and delivered, as applicable.

> Configuration Items (CIs) are the building blocks for a system, each uniquely identified and documented with its functional and physical characteristics. CIs provide the basis for applying management controls to a system configuration, permitting the isolation of items to be controlled, the tracking of status, and the reporting of configurations. CIs may take different forms:

- Software artifacts - the software CI is the architectural construct of the software system
- Hardware artifacts - the hardware CI is the architectural construct of the hardware

system

- Documentation artifacts - Documents, which describe CIs and configurations of CIs (baselines), may be CIs themselves

Note: The Configuration Management Office (CMO), in cooperation with systems engineering, establishes a scheme for the unique identification of each configuration item to be controlled for the Project/effort. Configuration baselines are the documentation that formally designate/identify the aggregation of CIs that constitute the system at any specific time. Configuration baselines, plus approved changes to those baselines, constitute the current approved configuration documentation. There are generally four formally designated baselines in the development life-cycle of a system, namely the Functional, Allocated/Design, Engineering, and Operational/Product baselines. Maintenance activities apply changes to the previous Operational/Product baseline to create the current Operational/Product baseline. Like named baselines are distinguished from one another through assignment of unique version designators. Additional information concerning configuration identification is contained in Section 5.2.

The component completes with a System Design Review (SDR), which determines whether the system is ready to proceed to the next Component in the SDLCMM process, the Engineer the Solution Component. Successful completion of Component 3 constitutes the Design/Allocated baseline. The start of planning for integration and deployment, and generation of the TIP, should also have been accomplished in this Component. Overall Project Manager, Executive Sponsor and Technical Project Manager review, approval, and sign off (Go/No Go Decision) for all disciplines at all appropriate levels is required at the SDR.

In Component 3, focus is directed toward:

- The organization/design of the system
- Definition of the data and associated business rules
- Definition of the processes
- The interaction among elements of the system
- Designing the system to meet performance, functional, and operational requirements
- Designing the system to meet reliability requirements
- Defining the cost and schedule for building, implementing and performing initial maintenance of the system
- The adaptability of the system to potential future requirements

These points constitute the realization of the system concept, which was defined earlier (the System Operations Concept, a Component 1 product) as a model of the system that defines the functional approach and the generic techniques that are to be used to achieve the prescribed performance. The portions of the system defined in the system design and the architectures proposed for them (COTS, hardware products, custom code, etc.) are examined together to achieve the best performance of the system. This component produces a system architecture, designed in detail and built as a set of independent portions/processes for subsequent integration into the completed, or updated/changed (for the maintenance case), system.

System Integration, Training, and Test Planning activities and the development and delivery of materials defined through these planning activities take place concurrently with System Design and continue to be addressed until full System Integration and Testing is completed in Component 5.

The Test Plan, which is begun in this Component, defines the planned test phases (i.e., unit test, software integration test, system test, user test, performance, systems integration test, acceptance test, etc.), assigns the verification tasks to the appropriate test phase, and lays out the organizational responsibilities in performing each test. These tests need to address all requirements defined for the system, both automated and manual. Further, they should ensure that user knowledge of the business area requirements are addressed if data value choices are not automated. It describes the test program at both high and low levels, levies requirements on the technical content of test plans and procedures, and establishes the criteria for verification of requirements.

The Test Plan must contain inputs from all disciplines that will be using and supporting the system. The Project Manager must ensure that this Test Plan does not represent only the input from the Project Technical Lead and his contractor. The Test Plan, by necessity, must include inputs from every concerned entity to ensure that all requirements are addressed in one or more tests and at all appropriate levels of test. Traceability of each system requirement to one or more tests must be documented. Review, understanding and approval is tantamount to ensuring a complete comprehensive testing program.

F.2.4 Component 4

The purpose of the Engineer the Solution Component is to move the allocated baseline of requirements and the (Component 3) design that was developed to address these requirements into reality by building, integrating, documenting, testing and preparing the solution for deployment. Component 4 takes a, possibly, preliminary design and carries it to the point where it is ready for final testing and user acceptance testing before deployment. Also included in this component’s effort are construction of all necessary policy, procedures and guidance for both the automated and non-automated aspects of the system addressing business knowledge and use of the automated functions which include preparation of the training materials to ensure:

- That the user understands how to use the application system
- That those who are responsible can operate the system

System artifacts and documentation are completed to ensure that the personnel who are designated to later support and service the application systems are provided sufficient information to perform their functions. As is the case with all components of the SDLCMM, NRC involvement and approval at all appropriate levels and at all appropriate times is required.

Formal documentation produced in this component may, as necessary, include updates to the schedule in the Project Management Plan (S-1052) and Software Development Plan (S-1057), updates to the System Operations Concept (S-3053), Test Plan (S-5151), Logical Design Document (S-3171), Physical Design Document (S-3172). During this Component, other critical documentation is written: Operational Support Guide (S-6151), User Training and Orientation Plan (S-7053), User Guide (6051), On-Line Help System and Tutorials (S-6052), Network Integration Diagrams (S-5056), entries are made to the Software Engineering Notebook (S-3091) and any necessary Rule Changes, Management Directives and business area functional guidance documents are developed and approved. This Component also includes extensive documentation work in the testing arena as the testing strategy gels. All planned levels of test must be planned and step-by-step procedures developed for conduct. Unit Testing, Application System Integration Testing, and Single Node Performance Testing is conducted. Test Reports, prepared post-test, are required to ensure traceability and communication that all system requirements, as baselined in the SRS, are satisfied.

Component 4 may be the first effort in the CPIC Project Phase but, at this point in the SDLCMM process, there are no special considerations for the CPIC process other than reporting back to the Overall Project Manager through the Technical Project Manager for the IT allocated portions and, as directed by the Overall Project Manager, for the manual approach portions of the allocated baseline. At a minimum, reporting consists of providing a status on whether the schedules and costs are within the bounds of the estimates and a projection on any effects to schedule and costs to ensure that problems are identified and addressed early. In this regard there are no differences between how this Component is viewed by a CPIC Project versus the Maintenance case.

The sequence of events in Component 4 is:

- Maintain the Change Log
- Design Validation (if not already completed)
- Design refinement (if not already completed)
- Code and Unit Test
- Integration and Test
- Create Rollout Strategy and Continue Deployment Planning
- Build the Users Manual
- Policies, Procedures and Training Materials Updated/Developed
- Update the Project Management Plan
- Review and Transfer Project Products to Central Configuration Management Organization

Each of these elements is accompanied by appropriate documentation and traceability to the system requirements.

F2.4.1 Maintain the Change Log

Configuration Change Control is the systematic proposal, justification, evaluation, coordination, approval or disapproval, and implementation of changes to Configuration Items (CIs), the planned discrete pieces of the system. The objective of Configuration Change Control is to preserve the integrity of controlled work products and the documentation associated with them. Configuration Change Control begins with the initial release of an item within a baseline, and continues throughout a product's life cycle including the operations and maintenance phases. Design changes are documented through the use of the Change Proposal Form (F-2502) and are logged in the Change Proposal Log Form (F-2561) as a record keeping and change management tool. It is the initiation and maintenance of that log that is the thrust of this activity.

Formal Configuration Change Control must have been established within each system to govern all changes and implemented at the Project level.

- No other method will be accepted for proposal of changes.
- "Formal" means "with a form" and with a specific review and approval process. The "form" may be electronic; the key is that there be auditable records of actual review and approval.
- Authority for change rests with the NRC; authority may be delegated, but responsibility cannot be delegated.

All requests must comply with defined change control requirements:

- Name and organization of requester
- Identification of configuration item/system to be changed
- Identification of other configuration items/systems which may be impacted
- Description of change
- Reason for change
- Date needed

Configuration Change Control processing is carried out through an aggregation of Problem Reports (PRs)/Change requests (CPs) and Configuration Control Boards (CCBs). PRs/CPs are completed to document all problems and changes. CCBs are convened to review perceived problems and proposed changes in order to decide if the problem is real or the change is needed, and to approve or disapprove proposed solutions.

F.2.4.2 Design Validation

This Section applies to projects/efforts for which the allocation of system requirements among CIs, design decisions and any associated downstream make/buy decisions are completely subject to the analysis and engineering judgment of the Project team, as approved by the NRC. In certain cases, however, some, or all, allocation, design and make/buy decisions have already been made earlier in the effort, either in the CPIC Selection Phase or in the CP/PR that sparked the Maintenance effort. Make/buy decisions are made in Component 1 and would necessitate a revisit of Component 1 by the Project Team should the need occur here.

In a development effort, once the transition is made from Component 3 to Component 4, prototyping, computer modeling and/or testing may be undertaken to determine if the best design concept was actually chosen in the Component 3 design effort. This ensures that the critical elements can meet operational and support requirements, and, if indicated, allows return to Component 3 for refinement of the design concept. In effect, this event would extend the scope of the CPIC Selection Process into Component 4 and necessitate the use of a Critical Design Review at the extended end of the design effort as the gateway to the CPIC Project Phase.

For the requirements allocated to a IT delivery approach, the purpose of the Design Validation step is to validate, if not already complete, a complete set of engineering requirements for each specific Configuration Item (CI). The requirements set includes functional, performance and general design (e.g., availability, safety, security, etc.) requirements as decomposed and allocated from the system-level requirements specification and derived and allocated from the system-level architectural design definition. These requirements are documented in the LDD, PDD and the Test Plan (TP) and any implications reported for entry into the SDP and PMP to be reported. A complete set of CI qualification requirements is also documented in the LDD/PDD. Activities during the Design Validation effort include:

- Validate for each CI the detailed engineering requirements allocated or derived from the SRS
- Validate for each CI the qualifications and methods needed to ensure that the CIs satisfy the requirements allocated to them
- Validate the requirements of any database to be used in common by more than one CI
- Validate the interface requirements for each interface external to each CI.

In the case of a Maintenance scenario, this activity may not be necessary unless a “design solution” accompanies the PR/CP that initiates a Maintenance effort. If a design is included in the PR/CP, then validation of that design, as described for a development effort, is required to the appropriate extent.

When present, Design Validation ends with a review, which may be formal or informal. It is prudent to plan to conduct reviews, as required during this effort, to ensure complete communication of Project and product status to all entities concerned. Most reviews in this context will be informal and in the form of Peer Reviews and/or Walkthroughs, but may include formal reviews as required or specified. When this effort is included in CPIC Selection Phase activities, the Critical Design Review (CDR) is the review gateway. All requirements, at all levels, must be reviewed by all appropriate organizational entities before being incorporated into the project.

For the requirements allocated to a manual approach, an analysis of the steps necessary to be taken to perform the work and deliver products that are called for (i.e., written policy, procedures, Management Directives, Training and Testing of Users in Business Principles, etc.) is conducted, assignments for work products are made and documented, and schedules are defined. Responsibility for creating this information rests with the Overall Project Manager.

F.2.4.3 Design Refinement

The Design Refinement effort is used to define the architecture and detailed design for each Configuration Item (CI) based on specified requirements, to develop designs for the interfaces specified and to partition the system into modules and define the design of each, to develop a design for any shared databases, and to develop a plan for testing the software against the requirements specific to the type of test and category of testing agreed to in the Software Development Plan and Project Management Plan. In certain cases, however, some, or all, design decisions have already been made earlier in the effort, either in Component 3, which is a part of the CPIC Selection Phase or in the CP/PR that sparked the Maintenance effort.

The detailed design effort is used to identify and design all the Computer Software Units (CSUs) that will comprise each Computer Software Component (CSC) within each Configuration Item (CI). The design is developed in sufficient detail to permit coding of the individual CSUs (i.e., programs, scripts, etc.). In addition to the design activities, CSU and CSC test requirements, responsibilities and schedules are defined (unit and process stream/sub-system testing). Test cases are defined for the informal CSU and CSC tests by developer/integrator. Test cases are developed for each of the formal CI tests identified in the Test Plan (TP) by the integrator. User documentation (i.e., User Guide, User Training and Orientation Plan, Operational Support Guide, Tutorials, etc.) are updated by Users with new information developed during this effort. Also during this step, the data base to support code and debug activities is allocated so that the data base can be certified ready for Code and Unit Test. Data bas allocation is as defined in the Data Models and is placed on the Development Test servers. In addition, the design of engineering software and procedures required to populate the production data base for delivery is developed.

During this effort, all required certifications for user business knowledge testing and full system testing scenarios (written from a business process perspective) are developed and documented. Additionally, training criteria for business area knowledge testing, new procedures and policy testing, etc. are defined and training programs and tests are are developed to address this criteria. Further, during this effort Business Advocates and Stakeholders define test scenarios and test data for the full system and the Overall Project Manager ensures that Acceptance Test scenarios and data are defined, matching the criteria defined in Component 1. The Overall Project Manager, or his designee, has the responsibility to ensure that the documentation, test data, and training materials cited in this paragraph are completed.

In the case of a Maintenance scenario, this activity will entail generation of the “design” of the solution or refinement of the “design solution” that may have accompanied the PR/CP that initiated the Maintenance effort. The design, in any case, must be integrated with the existing system to ensure that it introduces no adverse effects into the system. If a design is included in the PR/CP, then refinement of that design, as described for a development effort, is required to the appropriate extent.

When present, Design Refinement ends with a review, which may be formal or informal. It is prudent to plan to conduct reviews, as required during this effort, to ensure complete communication of Project and product status to all entities concerned. Most reviews in this context will be informal and in the form of Peer Reviews and/or Walkthroughs, but may include formal reviews as required or specified. When this activity is included in CPIC Selection Phase, the System Design Review is the review gateway. All requirements, at all levels, must be reviewed by all appropriate organizational entities before being incorporated into the project.

F.2.4.4 Build the Solution

Build the Solution is a Code and Unit Test activity which implements and tests the software which reflects the implementation of the design developed during the Design Component (and applicable refinement) and approved at the System Design Review (SDR) and/or Critical Design Review (CDR). The purpose of the Code and Unit Test effort is to produce and test each CSU defined in the detailed design for the Configuration Item (CI). Each CSU is reviewed for compliance with its corresponding detailed design description and applicable coding standards prior to establishing internal configuration control of the CSU and releasing it for integration. Additionally portions of these activities include developing detailed procedures for conducting each informal CSU and Computer Software Component (CSC) integration test. CI and build test procedure generation is begun and will be completed during the CSC Integration and Testing effort. There is no formal review during Code and Unit Test, but it is prudent to plan to conduct reviews, as required, to ensure complete communication of Project and product status to all entities concerned. Most reviews in this context will be informal and in the form of Peer Reviews and/or Walkthroughs, but may include formal reviews as required or specified. The outputs from this effort include the source code listings, source code, and the Procedures portion of the testing documentation. This activity ends with the Test Readiness Review (TRR).

In a Maintenance scenario, implementation of the design may range from trivial to extremely complex. In any case, the appropriate level of coordination and planning for Code and Unit Test must be completed. In most cases, this will entail careful planning for coding and regression testing to ensure continued integrity of the system.

F.2.4.5 Integration and Test

The purpose of the Integration and Test activity is to integrate and test the application system software and is considered the first formal test. Testing here concentrates on verifying that the developed (implemented) CI meets the requirements specified in the SRS and is representative of the Concept of Operations defined by the Project Team in Component 1. Where applicable, CSC testing is performed to aggregate modules or units before aggregating CSCs for integration into CIs. Verification in this instance means to establish the correspondence between a software product and its specification. This testing process is performed during the development cycle and can begin as early as when the code is developed and may extend to the end of Integration and Test. It is complete when all CSCs and CIs have been successfully tested in accordance with the appropriate test plan (formal test scenarios and procedures).

Computer Software Component (CSC) Integration and Testing is used to perform a series of builds and tests to verify that a Configuration Item (CI) is complete and ready for formal testing by the full Project Team and the Users and Operations Support Staff who will be responsible for running the system in production. CSCs are aggregated using a cycle of build, build test and integration test. Builds, in this context, are aggregations of CSCs that represent a defined functionality, structure, or scenario and are sometimes the entry point for new Computer Software Units (CSU) into Project Configuration Management. Builds are generally performed by a dedicated build team using the baseline configuration contained in the CM library. Integration tests determine by functional, structural, statistical, and regression testing that the software performs correctly, generates expected results, and handles errors appropriately. The CSC Integration and Testing effort is iterative and may overlap Code and Unit Test. CSC Integration and Testing can begin as soon as groups of CSUs making up CSCs are available in the Development Configuration. The outputs from this activity include the tested Development Configuration and updated SENs for each CSC and CSU. Test reports document each test result.

In a Maintenance scenario, integration and testing of the changed portions of the system must include the appropriate level of coordination and planning to ensure continued integrity of the system.

When all Engineering effort is complete, with all entities integrated and successfully tested, Component 4 is ended. The Engineering Baseline is established at this time as the basis for System level and acceptance testing with the customer organization. Traceability of all system entities (CIs), and the testing thereof, to the system requirements must be maintained and documented.

F.2.4.6 Create Rollout Strategy and Continue Deployment Planning

The purpose of this activity is for engineering, program and/or operations personnel who are involved in the building and using of a system to describe the integration of the hardware and software components that comprise the system for final deployment in the customer's environment. It entails researching and deciding how, where and when the system will be deployed, what the precedent actions are with respect to the environment and creating a strategy for putting the system in place.

After determination of the target user community, an assessment of the software/hardware platforms, configurations and environments will reveal the need for upgrade, customization and/or replacement. Those things considered, a schedule for preparation of the environment can be completed before actual installation and rollout of the system. The results of this planning effort are documented in an update of the Tactical Implementation Plan.

In a Maintenance scenario, it is most important to maintain continuity between the existing and the new system in terms of operability. To achieve that end, emphasis must be placed on interface with the existing users to ensure that system, operational and environment changes are communicated and that appropriate training is conducted in advance of rollout and/or deployment.

F.2.4.7 Build the Users Manual

The Build the User Manual activity encompasses not only putting together the actual User Guide, but assembling the User Training and Orientation Plan and developing a Customer Support Center.

The User Guide is targeted for personnel who actually use the software, i.e., execute the application, to perform their job. The User Guide is both a training and a reference resource. The effort that results in production of the User Guide also produces the Help System, Operational Guide, and Tutorials.

Knowledge of the target operational system and the requirements to support that system are melded together to define the Customer Support Center. The Customer Support Center has responsibility for servicing the user after training as s/he uses the system in the true operational setting.

Overall training plans and strategies are developed from knowledge of the operational system, the known skill set of the user and other training materials. From this information, the User Training and Orientation Plan is assembled to assess the skills of the users, to tailor a training program for the target users, to evaluate resource needs to effect the training and to assess the effectiveness of training. Update of User Manuals, training and interface are most important in the case of a Maintenance effort.

F.2.4.8 Policies, Procedures and Training Materials Updated/Developed

After an analysis of the NRC's current policies and directives related to the business areas and processes supported by the application system design, develop or update to ensure that they support the business processes and rules as envisioned in the design and Systems Operational Concept. It is important at this point to begin the development of user training materials for use in actual training. Training should address both the manual and automated process that make up the system, as well as business knowledge of the user. This Activity ensures that all policies and procedures that are the basis for work flows and business rules defined to the application system and presented in user training are in fact updated and ready for publication. Once training materials are developed, actual training of business process users may begin in both the automated and manual operations arenas for both development and maintenance efforts.

Updated policies and procedures will be finalized, published and fully implemented prior to deployment in Component 5.

F.2.4.9 Update the Project Management Plan

Based on the activities in this component and the documents generated, it is important to now go back and update the PMP in preparation for Component 5 activities. The documents that become inputs for the PMP update are shown in Activity 4-8.

F.2.4.10 Review and Transfer Project Products to Central Configuration Management Organization

Review the results of the activities that were performed in this Component to determine if the system should be moved forward to Pre-Deployment/Final Testing. Subject Matter Experts in the area of Records Management and the Infrastructure, along with the Technical and Overall Project Managers, should be involved in this review activity to ensure that all aspects of the system and the data are reviewed and commented on from those unique subject perspectives and that the system is acceptable to move forward. If approved (Go/No Go Form), all the release artifacts are delivered to the Central Configuration Management Organization to audit and check-in prior to start-up of Component 5 activities.

F.2.5 Component 5

The purpose of the Deploy the Solution Component is move the baselined product into production. This requires ensuring that the product is ready for deployment, the user knows how to use the system, the operator knows how to care for the system, and the environment is ready for the system. This is accomplished by performing system level testing (includes integration and operational performance testing such as threshold and throughput), user testing, and acceptance testing (user and operational support) before actual deployment of the application/system. It includes delivery of the training to:

- Ensure that the user understands how to use the automated application system and is proficient in the business rules which defines the way in which the application system is used.
- Operator support training which ensures that such actions as backup, recovery, software installation, and user access are understood and Operations Support Guides are reviewed for accuracy prior to deployment.

It also includes, where necessary, making upgrades to the infrastructure environment and transferring data from old systems. It ends with the actual deployment of the new system.

As is the case with all components of the SDLCMM, NRC involvement and approval at all appropriate levels and at all appropriate times is required. However, unlike other Components where a single Go/No Go decision is made, this Component requires two:

- A decision on deployment
- A decision on whether the development team will maintain the deployed version of the system for a set period of time, usually resulting from an approach that calls for a phased delivery of the system, or it the application will be transferred to the maintenance contractor, which means a Go/No Go decision to Component 6. (This decision is moot in the case of a maintenance effort since the system is already in maintenance.)

Formal documentation produced in this component includes preparation of the product deployment announcement, updates to any of the Plans and other documents, and extensive documentation work in reporting test results and update of the technical inventory. All planned levels of test must be executed using the step-by-step procedures developed for their conduct, as documented in the Test Plan and scenarios. Post-test Test Reports are required to ensure traceability and communication that all system requirements, as baselined in the SRS, are satisfied.

> Note: In general, this section describes the process that is followed when executing a CPIC initiated effort. A maintenance effort requires evaluation at each step to tailor the effort for the particulars of the intended release. There is no less rigor employed in the maintenance effort, but there is also no need to expend unnecessary effort for a less complex release. This section describes the full process, with notes and explanations for consideration in tailoring the process for a maintenance effort.

The sequence of events in Component 5 is:

- Review and Update of Plans and Announcement of Deployment
- Validation and Upgrade of the Operational Testing Environment
- Installation, System Integration Test and Evaluation
- Operational Support System Level Training, Test, and Evaluation
- Implement Policies and Procedures
- Training of the Users
- User Testing and Evaluation
- Acceptance Test and Evaluation
- Validation and Upgrade of the Production Environment
- Cutover and Deployment
- Update of Technical Inventory

Each of these elements is accompanied by appropriate documentation and traceability to the system requirements.

F.2.5.1 Review and Update of Plans and Announcement of Deployment

Review the Project Management Plan and TIP, and prepare announcement for the upcoming deployment. The announcement should address information that ensures that all concerned personnel are aware of the implementation of the new system capabilities/changes and that each person (user, operator, manager, etc.) understands the overall and individual impact. It is important to prepare people to expect deployment activities. See Activity 5.1. This announcement will be deployed to all interested and/or concerned parties in accordance with the schedule defined in the Project Management Plan, which has a schedule dependency linked to satisfactory Acceptance Testing (Activity 5.12).

F.2.5.2 Validation and Upgrade of the Operational Testing Environment

Care must be taken at this point to ensure that each of the testing environments is conducive to the testing activity which it must support. Validate each environment and make necessary changes to ensure successful testing prior to deployment. See Activity 5.2.

F.2.5.3 Installation, Application System Integration Test and Evaluation

This activity is geared more toward a full (CPIC) development effort, but is important, to the extent applicable, for a maintenance effort in that the correct level of integration is accomplished to ensure that changed elements do not adversely affect the system. This action sets the stage for regression testing in the maintenance case.

The Application System CIs are installed in the proposed operational environment. Application System Integration Test and Evaluation activity takes place after the CIs (discrete portions of the application system) have been integrated and tested. Application System Integration and Test encompasses software, hardware and firmware. This activity is concerned with the integration and testing of the CIs to ensure that the interfaces between CIs, or combinations of them that make up sub-systems, are properly connected and to verify those requirements that can not be verified without an integrated system.

The purpose of the Application System Integration Test and Evaluation activity is to integrate the aggregate CIs into the end product system by conducting subsystem level integration of CIs into CI subsystems and integrated subsystems against the requirements of the SRS, then integrating subsystems into system segments or into a full system configuration. Test and audit of the system is conducted for responsiveness to documented functional and physical requirements. Various test scenarios, including end-to-end testing scenarios and, as applicable, integration testing should be utilized to test the Application System processes and functionality.

Application System Integration Testing activities will occur in the Consolidated Test Facility (CTF) which is a "lab type" integration facility. Inputs to this activity are the applicable Test Plan sections, including step-by-step procedures for each intended scenario. The output of this activity is the System Test Report (STR). See Activities 5.3, 5.4 and 5.5.

F.2.5.4 Operational Support System Level Test and Evaluation

Operational Support System Level Testing and Evaluation entails activities that are required to conduct technical product acceptance testing from an operation support perspective. This type of testing is critical to perform prior to user acceptance and deployment of the application system. This is where the final technical verification of the application system is performed. Operational Support personnel are instructed on the application system and its components and operational support requirements. These personnel are provided with an Operational Support Guide which addresses such tasks as backup, recovery, and installation. Verification addresses testing against system performance, operational, and security requirements. Throughput and threshold tests are performed. The application systems ability to address requirements for security, backup and recovery, and installation into the infrastructure are tested and evaluated. Further, there is a level of full infrastructure integration (cohabitation) testing conducted. An evaluation of document usefulness and accuracy is performed against the Operational Support Guide, the Installation Instructions, etc. At the end of this testing and evaluation, all components of the application system product/operational baseline (documents, code, executables, integrations, software and hardware) are expected to be ready for deployment.

F.2.5.5 Implement Policies and Procedures

This Activity centers on ensuring that all policies and procedures identified and updated in Component 4, which are the basis for work flows and business rules defined to the application system and presented in user training, are, in fact, finalized, published and implemented.

F.2.5.6 Training of the Users

Users must be thoroughly trained in the use of the application system and possess a certified level of knowledge regarding the business area supported by the application system. Training plan and training materials developed earlier in the SDLCMM are used to schedule and present this training. Certification testing may be conducted to ensure that Users are knowledgeable in the business areas and appropriately familiar or trained in the Policies and Procedures developed in Activity 5.8.

F.2.5.7 User Testing and Evaluation

Testing of the application system from a user perspective is conducted using both users certified in the business area and those who are not. Validation of the adequacy and ease of use of the Policy and Procedures, User Guide and On-Line Help and Tutorials is conducted and results are recorded and evaluated. Improvements to training, policy, procedures and the application systems are captured through a Change Proposal and Problem Report approach and evaluated along with test results.

F.2.5.8 Acceptance Test and Evaluation

Using the criteria defined by the Sponsor for acceptance of the application system and captured as scenarios to the Test Plan, the Project Sponsor and his designees conduct the Acceptance Test against the technically accepted application system. Results of these test scenarios are captured and evaluated as part of the User Satisfaction Determination and the Go/No Go Decision. Documentation of these decisions must be completed after all of these levels of testing have been conducted and there has been a determination of resolution for outstanding issues. With the Go/No Go Decision made to move forward, the cut-over and deployment activities can be pursued.

At the end of these activities, test results and any updated documents are gathered for delivery to CCM for use in completing the conduct of the final Functional and Physical Audits prior to deployment. This activity is completed after successful conduct of the Cutover Readiness Review (CRR). The first Go/No Go decision is made at this point and documented by approval of the Solutions Modules Ready for Deployment Form (F - 4051). The announcement of deployment can now be published. All testing information is delivered to the Central CM Organization for their use in completing the Functional Configuration Audit. The Central CM Organization provides the necessary artifacts for the deployment to the appropriate parties after these audits are completed and the audits are favorable for deployment.

F.2.5.9 Validation and Upgrade of Production Environment, Cutover and Deployment of Application System

The purpose of the Validation and Upgrade of Production Environment, Cutover and Deployment of Application System Activities are to take any necessary steps needed to prepare the production environment (installing upgrades and vendor production) and to prepare prior application systems data or copy control tables, etc. for use in the new application system or release, and to release the application system, as supplied by the Central CM Organization, into the production environment. The observation of its performance in actual operation, and generation of necessary changes through the CM change control process continue into Component 6 activity or into continued maintenance activities under the development contract. CPIC Projects require that a report be submitted on Lessons Learned after 6 months of operation.

This Component ends with the delivery of the product into production. Detailed guidance for the conduct of Deployment activity is contained in Activity 5.15. Activity context for Production and Deployment is shown in Figure 10-1

If maintenance of the application system is transferred to another contractor or contract under the second Go/No Go decision, the Project shall be closed out in accordance with appropriate close-out procedures or as specified in Project Management directives.

Appendix G. Glossary

Acquire Support Resources. SDLCM Methodology Component 2, whose purpose is to obtain the necessary resources to ensure timely and effective progress on the Project.

Activity. A unit of work that has well-defined entry and exit criteria. Activities can usually be broken into discrete steps.

Application System. A discrete set of information resources organized for the collection, storage, processing, retrieval, maintenance, use, sharing, dissemination, or disposition of information.

Baseline. A document or set of documents, code, executables, etc. that makes up a system. After a baseline is established, the Nuclear Regulatory Commission (NRC) must approve all changes prior to their incorporation. Three discrete baselines exist in a Development Project: Requirements/Functional, Design/Allocated, and Operational. Operational baselines are surrendered to Central CM prior to testing.

Build. A version of software that meets a specified subset of the requirements that the completed software will meet. (See also *release*.)

Component. **1.** One of the seven structural divisions of the SDLCM Methodology. **2.** A module (See *module*.)

Computer software component (CSC). A logical subdivision of a computer software configuration item (CSCI). CSCs are a distinct part of a computer software configuration item. CSCs may be further decomposed into other CSCs and to yet smaller computer software units (CSUs).

Conceptual data model. One of the three data models; identifies groupings of data important to the business situation that the Project addresses. (See also *logical data model*, *physical data model*.) The Conceptual data model represents the level of detailed knowledge of system requirements and design known prior to the completion of Component 3. The level of knowledge known at the completion of the Design (end of Component 3) is reflected in the Logical data model and a preliminary Physical Data Model. The Conceptual data model is a preliminary Logical data model.

Conceptual process model. One of the two process models; also known as the scope-setting version of the logical process model. (See *context diagram*. See also *logical process model*.) The Conceptual process model represents the level of detailed knowledge of system requirements and design known prior to beginning the design activities in Component 3. The level of knowledge known at the completion of the Design (end of Component 3) is reflected in the Logical Process Model. The Conceptual process model is a high-level, preliminary Logical Process Model.

Configuration audit. The mechanism for verifying the system configuration. Configuration audits should address the examination of the system against the functional requirements and the SDLCMM required products (physical). Configuration audits are conducted on the system at both the Project Level and Central Configuration Library Level by the Configuration Management personnel.

Configuration identification. The application of a formal set of procedures to select CIs, determine the documentation required for each CI, affix unique identifiers to the documentation that defines the CI, release CIs and their associated documentation, and establish configuration baselines for each CI.

Configuration item (CI). System component (for example, hardware CI or software CI) that is developed or purchased, controlled, accepted, and maintained separately from other system components. In practice, it is a component that is convenient and sensible to document and control as an entity. (See *software configuration item*.)

Configuration management (CM). The application of a formal set of procedural concepts by which a uniform system of configuration identification, control, authentication, and status accounting is established and maintained for a system. These procedures are documented for a Project through the development of the Configuration Management Plan.

Configuration status accounting. The recording, monitoring, and reporting of all changes to established baselines for each CI. The purpose of configuration status accounting is to record and report all information needed to manage configuration items effectively. The objective of configuration status accounting is to ensure that all approved changes for a CI are reflected in associated baselines that describe the CI and that no baseline includes any changes that have not been approved.

Context diagram. A top-level data flow diagram 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. (See *conceptual process model, data flow diagram*.)

COTS (or **GOTS) software.** Commercial (or government), off-the-shelf software. COTS and GOTS software includes (1) unique components that must be delivered with the product to execute the operational software and (2) development tools that must be delivered with the product to support maintenance of the software.

CPIC - Capital Planning and Investment Control. A policy of the NRC to ensure that information resource investments are planned, selected, managed and evaluated to maximize the value and minimize the risks of those investments, in accordance with Federal statutes and regulations (M.D. 2.2)

Data flow diagram. A logical graphical representation of the data that are processed and generated during system operations.

Data model. (See *conceptual data model*, *logical data model*, and *physical data model*.)

Decommission The Solution. SDLCM Methodology Component 7, whose purpose is to inactivate and cease support of an application by appropriately addressing the Records Disposition Schedule conditions of the system.

Define Initial Project Requirements. SDLCM Methodology Component 1, whose purpose is to identify an information management problem, clarify scope, identify functional and data requirements, analyze alternative solutions, select appropriate tools, establish a Project Management Plan, and develop a support resource request.

Deploy the Solution. SDLCM Methodology Component 6, whose purpose is to implement and roll out the integrated solution.

Design. Those characteristics of a system, including its associated hardware and software CIs that respond to the requirements (see *requirements*). Some will match the requirements; others will be elaborations of requirements, such as definitions of all error messages in response to a requirement to display error messages; still others will be rules, policies and procedures for support of manual data and process actions, including identification, collection, etc.; decisions on how processes and data logic will be used, etc. to satisfy the requirements.

Design The Solution. SDLCM Methodology Component 3, whose purpose is to determine functional, data, and communications or network requirements, document user interface requirements, design the solution, develop necessary enabling rules, procedures, policy, directives, etc. and prepare integration and Project plans.

Development. Production of a new product that leads to delivery of all initially planned functional capability. The new product may be built from newly created components, reused components (with or without adaptations), GOTS components, and COTS components. (Contrast with *enhancement, maintenance*.)

Deviation. A departure from a requirement, an alteration of an activity defined by a procedure, or a variation in the production of a product defined by a standard. (Contrast with *waiver*.)

Documentation. A collection of data, regardless of the medium (or media) on which it is recorded, that generally has permanence and can be read by humans or machines. The significance of this definition is that documents do not necessarily have to be separately bound entities; they may comprise data in several media (for example: plans, data models, outputs from CASE tools, databases).

Effort. A defined and structured application of the SDLCMM to respond to maintenance change proposals or emergency fixes. The word *effort* is differentiated from the word *Project* in that a unique reportable schedule and budget are not subject to CPIC review, report and oversight.

Engineer the Solution. SDLCM Methodology Component 4, whose purpose is to construct or acquire all modules of the system, perform tests, create the rollout strategy, and construct the training and education materials

Engineering change. A change to the currently approved configuration documentation of a configuration item at any point in the life cycle of the item. Generally, these are documented on a Change Proposal and approved through an established Project-level Change Control Board.

Enhancement. A major addition or change in the functionality of an operational system; often includes many of the same activities as new development. Generally, reported to and controlled through the CPIC process. (Contrast with *new development, maintenance;* see also *software engineering*.) Minor enhancements are treated as perfective maintenance.

GOTS software. (See *COTS software*.)

Inspection. A process whereby products are reviewed for correctness, completeness, quality, and compliance with requirements and standards. The process is carried out by one or more peers of the product's developer. (Contrast with *walkthrough*.) Inspections, their frequency and responsibility for conducting them are documented in the Project's Quality Assurance Plan (QAP).

Joint application design (JAD). A facilitated workshop technique that brings together principal stakeholders to solve a well-defined problem to produce a well-defined product or set of products.

Life cycle. (See *system life cycle.*)

Life-cycle model. A framework on which to map activities, methods, standards, procedures, tools, products, and services (for example, waterfall, spiral).

Logical data model. One of the three data models; the completed version that presents a fully attributed and normalized data model and identifies the relationships among all data entities. (See also *conceptual data model*, *physical data model*.) The Logical Data Model improves upon the Conceptual data model by further defining the data model based on the allocated data requirements. The Logical Data Model represents the data in a model which reflects the understanding of the data and its use, from a design standpoint, at the close of Component 3.

Logical process model. One of the two process models; the completed version that provides all the details required to document and communicate to the business community the Project team's understanding of the functional requirements of the application system. (See also *conceptual process model*.) The Logical Process Model represents the level of detailed knowledge of system requirements and design known prior to completing the design activities in Component 3. The level of knowledge known at the completion of the Design (end of Component 3) is reflected in the delivered system.

Maintenance. Implementation of problem fixes and minor enhancements to an operational product, also called Servicing the Solution and documented in the SDLCMM under Component 6. (Contrast with *new development, enhancement*)

Method. A technique or approach, possibly supported by procedures and standards, that establishes a way of performing activities and arriving at a desired result.

Methodology. "A collection of methods, procedures, and standards that defines an integrated synthesis of engineering approaches to the development of a product." [Paulk]

Milestone review. A process or meeting involving representatives of the Project team and the parties assigned specific roles which produce products at which Project status, product status, and both Project and product issues are examined and discussed.

Module. A cohesive group of software units that performs a software function.

Non-developed item (NDI). A component that is not newly created by the software team. This includes reused components, COTS and GOTS components, and components developed by other government groups or contractor personnel.

Physical data model. One of the three data models; the design model, which describes how data will be distributed to different processing nodes and structured to meet performance objectives in a specific physical implementation. (See also *conceptual data model*, *logical data model*.)

Procedure. A written description of the roles, responsibilities, and steps required for performing an activity or a subset of an activity. (Contrast with *standard*.)

Process. "A sequence of steps performed for a given purpose; for example, the software engineering process." [IEEE 610.12]

Process Model. (See *conceptual process model, logical process model*.)

Product. Software, procedures, rules, policy/directives, models, documentation and other associated information created, modified or incorporated to satisfy the requirements of a Project or Effort. (Examples include plans, requirements, design, code, databases, test information and manuals.)

Project. A resource investment that is subject to the CPIC controls. A Project consists of the work to be done and the personnel assigned to perform the work. Products are produced within a Project.

Prototype. An early experimental model of a system, system component, or system function that contains enough capabilities for it to be used to establish or refine requirements, or to validate design concepts.

Qualification testing. Testing performed to verify and demonstrate (often to the customer) that a software CI or a system meets its specified requirements.

Release. 1. A build that is delivered to the customer. (See *build*.) **2.** An action performed by the configuration management organization making a particular version of software available for a specific purpose (for example, for independent testing, or for installation into the production environment).

Requirement. A characteristic that is initially defined in Component 1 which defines a specific need for a system to deliver a capability, operational function, programmatic function or data.

Service The Solution. SDLCM Methodology Component 6, whose purpose is to maintain, enhance, or modify an existing application or its support environment.

Software engineering process. An organized set of activities performed to translate requirements and operational concepts into software products.

Software engineering. A set of activities that results in software products. Software engineering includes new development, modification, reuse, reengineering, maintenance, or any other activities that result in software products.

Software product. Software or associated information created, modified, or incorporated to satisfy the software requirements. Examples include plans, requirements, design, code, databases, test information, and manuals.

Software product evaluation. Activities performed by the software team to ensure that in-process and final software products meet criteria established for those products.

Software Development team. The group, led by the Technical Project Manager, responsible for engineering software products that automate specific system requirements (including new development, modification, reuse, reengineering, maintenance, or any other activity that results in software products).

Software test environment. The facilities, hardware, software, firmware, procedures, and documentation needed to perform qualification, and possibly other, testing of software. Elements may include but are not limited to simulators, code analyzers, test plan generators, and path analyzers, and may also include elements used in the software engineering environment.

System transition. The set of activities that enables responsibility for system engineering to pass from one organization, usually the organization that performs system development, to another, usually the organization that will perform operational support and maintenance.

Standard. Written criteria used to develop and evaluate a product or to provide and evaluate a service.

System life cycle. “The period of time that begins when a system is conceived and ends when the system has been decommissioned.

Timebox. A subdivision of a release created to achieve a unit of production that is manageable in size, complexity, staffing, and duration. It is a planning construct that controls functionality delivered by fixing resources and time.

Version. An identified, documented, and controlled collection of automated and manual processes that make up a system, its associated data structures, procedures, policies and documentation. Any change to a version and its associated documentation and data structures requires a configuration management action.

Waiver. The elimination of a specific requirement to perform an activity or to produce a product. (Contrast with *deviation*.)

Walkthrough. Detailed technical presentation of a limited aspect or portion of a product. Such presentations are usually made by the product’s developer.

Acronyms

AAD	Alternatives Analysis Document
BPR	Business Process Reengineering
CCB	Configuration Control Board
CCM	Central Configuration Management
CDR	Critical Design Review
CI	Configuration Item
CIO	Chief Information Officer
CLIN	Contract Line Item
CM	Configuration Management
CMO	Configuration Management Office
CMP	Configuration Management Plan
COTS	Commercial Off-The-Shelf
CP	Change Proposal
CPIC	Capital Planning and Investment Control
CSAD	Current System Assessment Document
CSC	Computer Software Component
CSCI	Computer Software Configuration Item
CSU	Computer Software Unit
DFD	Data Flow Diagram
DM	Data Management
FCA	Functional Configuration Audit
GILS	Government Information Locator System
GOTS	Government Off-The-Shelf
GSA	General Services Administration
IRM	Information Resources Management
IT	Information Technology
JAD	Joint Application Design
LDD	Logical Design Document
NDI	Non-Developed Item
NRC	Nuclear Regulatory Commission
OCIO	Office of Chief Information Officer
OIRM	Office of Information Resources Management
PAL	Process Assets Library
PCA	Physical Configuration Audit
PCM	Project Configuration Management
PDD	Physical Design Document
PFD	Process Flow Diagram

PMP	Project Management Plan
PR	Problem Report
QA	Quality Assurance
QAP	Quality Assurance Plan
RFPA	Request for Proposal Action
RM	Records Management
SDIB	Systems Development and Integration Branch
SDLCM	System Development and Life-Cycle Management
SDLCMM	System Development and Life-Cycle Management Methodology
SDP	Software Development Plan
SDR	System Design Review
SEN	Software Engineering Notebook
SME	Subject Matter Expert
SOC	System Operations Concept
SOW	Statement of Work
SQA	Software Quality Assurance
SRS	System Requirements Specification
SSR	Software Specification Review
TBD	To Be Determined
TIP	Tactical Integration Plan
WBS	Work Breakdown Structure

References

[IEEE 610.12] ANSI/IEEE-STD-610.12-1990, “IEEE Standard Glossary of Software Engineering Terminology,” February 1991.

[ISO/SEC 12207] ISO/SEC 12207, “Software Life-Cycle Processes.”

[MIL-STD-498] MIL-STD-498, “Software Development and Documentation.

[NRC MD2.5] NRC Management Directive 2.5, “Application Systems Life-Cycle Management,” Draft, January 1997.

[NRC SH1.1] NRC OIRM/SDIB, *SDLCM Methodology Handbook*, Version 1.1, November 1996.

[NRC SH2.0] NRC OCIO/ADD, *SDLCM Methodology Handbook*, Version 2.0, December 1997.

[NRC SH2.2] NRC OCIO/ADD, *SDLCM Methodology Handbook*, Version 2.2, December 1999.

[Paulk] Paulk, M., et al., *Key Practices of the Capability Maturity Model, Version 1.1,* Software Engineering Institute, Carnegie Mellon University, CMU/SEI-93-TR-25, February 1993.