NATIONAL ARCHIVES and RECORDS ADMINISTRATION

INFORMATION TECHNOLOGY

Systems Development Lifecycle Handbook

Supplement to NARA 805

July 8, 2005
Version: 1.3

Table of Contents:

List of Figures

List of Tables

1 Introduction

1.1 Background and Purpose

In response to statute and OMB guidance, the National Archives and Records Administration (NARA) Director of Human Resources and Information Services (NH) has undertaken the building of an Enterprise Systems Architecture and Information Technology infrastructure. The goals are to document and improve the automation of business processes and to improve process and practices employed in building information systems.

NH has developed this NARA Systems Development Lifecycle Handbook as a supplement to the policy directive, NARA 805. The handbook presents a set of processes and procedures to be followed by project teams during the development, evaluation, and maintenance of NARA Information Technology (IT) systems. It has been created to

- support the IT Product Owner in understanding the scope of IT activity;
- support the IT Project Manager's efforts in planning the schedule and tasks necessary to successfully implement an IT system;
- assist the development team in understanding process expectations; and
- assist Government reviewers in validation of work products.

This document has been written with the current NARA organization in mind. Specifically, this assumes that NARA is primarily an acquirer of IT systems rather than a developer; however, the IT development process model can be used towards in-house development of systems and applications when an appropriate need exists.

In the development of this lifecycle process for NARA IT systems there are several guiding principles that have been used in its creation. To wit;

- Recognize Government emphasis on commercial-off-the-shelf (COTS) integration in place of custom development. COTS based systems often have a significant impact on the flexibility of implemented solutions, but it is a reality of the marketplace that COTS based systems are needed to help limit costs associated with systems development. NARA is expected to follow a COTS based approach in the majority of its systems and will reflect this expectation in the lifecycle tasks. However, the process must accommodate custom development when required.

- Continuous risk management is applied to the process. Techniques such as frequent user involvement and reviews, prototypes and pilots, and applying varying levels of rigor to the process and products are all used. Involving the users of the system in its development is key and must be facilitated to the extent possible. Ensure security is incorporated into all phases, from concept development to retirement.

- The process must be adaptive (or flexible) enough to work with a variety of systems development approaches as applicable to various stages of the lifecycle, such as Rapid Application Development (RAD) Rational Unified Process (RUP), Extreme Programming, traditional structured analysis and object-oriented approaches. The lifecycle process should not be viewed as encouraging any one form of development methodology over another, but rather an attempt to capture the best practices from those that are well known and likely to be used in NARA IT systems. During the systems acquisition phase, NARA will formally evaluate each proposed methodology (other than MED) against a set of criteria that ensures a consistent cross-walk between what NARA requires to comply with the SDLC in all its phases and another methodology's proposed processes and deliverables. Alternative system development methodologies will be approved by the Director of the Systems Development Division.

Each of these principles and the associated practices are explored and developed in later sections of this document.

1.2 Scope

This document defines the NARA Systems Development Lifecycle (SDLC) model. Processes covered by the SDLC model consist of analysis, design, construction, and implementation of IT systems along with management of those tasks. Guidelines pertaining to other areas of NARA product planning and management efforts such as product approval, acquisition planning, business process change management, etc. are contained in other documents. Some aspects are referenced in this document when they are closely related to the systems development process.

Assumptions have been established that set a boundary on the scope of this document. It is assumed that activities in this document take place after

- A statement of need has been developed
- The product planning effort is complete
- The project has been approved for development

Essentially, it defines activities after a project has been approved and project execution is ready to take place. However, it should also be used in the "Product Planning" process to help determine the level of effort required in completing a project based on project complexity, size, and risk.

1.3 Document Layout

This document is separated into two levels of information. The first, section 2, provides an overview of the Managed Evolutionary Development (MED) process to familiarize readers with the preferred NARA development lifecycle approach. The second level provides specific details relating to MED activities. The sections (and activities) are

- Concept Development (section 3)
- Requirements Definition (section 4)
- Preliminary Design (section 5)
- Detailed Design (section 6)
- Development (section 7)
- Integration and System Test (section 8)
- Deployment and Acceptance (section 9)
- Production (section 10)
- Retirement and Rollover (section 11)

Each major MED activity is provided its own subsection that includes

- Task Outlines;
- Review Checklists; and
- Work Product or Document Templates.

1.3.1 Task Outlines

The Task Outlines provide a comprehensive set of tasks required during each build activity of a system. They are intended to let the project team know *what* must be done, not *how* to do them. (To learn more about how these tasks may be performed, see section 13 - Selected Readings at the end of this document.)

Each Task Outline included in the NARA systems development lifecycle has the following layout:

Purpose -	Describes the intended outcome of the activity and its tasks.
Timeline Diagram -	Process flow diagram intended to show the *usual* sequence of tasks and any relationships that exist between them. These diagrams also show which tasks are responsible for generating a work product. Any individual project may be more or less complex than that shown and the linkages between tasks should be maintained via the schedule and illustrated through Gantt charts or other mechanisms.
Entry Criteria -	Inputs that must be provided or Actions that must have occurred prior to starting the activity.
Work Products -	The expected documentation or system components produced during the activity.
Exit Criteria -	Outputs that must be produced or Actions that must occur before the activity is considered complete.

Engineering Tasks -	Specific tasks that are intended to produce system components or related documentation (and are maintained minimally for the life of the *system*). Each primary engineering task is numbered for cross-referencing with the Activity Timeline Diagrams and the Roles / Responsibilities section. Tasks are identified as E1, E2, ... En in the expected sequence.
Management Tasks -	Specific tasks that are intended to produce items related to how the engineering work is performed (and are maintained minimally for the life of the *project*). Each primary management task is numbered for cross-referencing with the Activity Timeline Diagrams and the Roles / Responsibilities section. Tasks are identified as M1, M2, ... Mn in the expected sequence.
Roles / Responsibilities -	Describes who has primary responsibility for a given task or needs to participate in a support capacity. Projects may be staffed such that individuals serve in multiple roles.

1.3.2 Review Checklists

1.3.2.1 Format

The review packages and checklists are provided to give both the project team and reviewers a foundation for conducting milestone reviews in conjunction with the Task Outlines. They are comprehensive in nature and may require adjustments based on task size, complexity, and risk assessments made during project planning.

The review checklists include factors that should ensure a base level of quality in work products. Where possible, these factors are written to be objective evaluators of the work and can be used by the development team as additional guidelines for project deliverables. Of necessity, the factors are generic in nature. When appropriate, the quality assurance and development team should refine these items so that they are specific to the project.

1.3.2.2 Usage

Each review process has a package to define the purpose and objectives, who should present, who should review, when it should be done, key issues, entry and exit criteria, agendas, and review checklists.

The "key issues" for each review tend to be subjective and have broad scope. In some cases they are also open-ended rather than just yes or no. These questions may prompt discussion during the reviews, but the review team should reach a consensus that the key issues have been adequately addressed for the review to be successful.

Checklist questions, on the other hand, are written to evoke a yes / no response. For successful completion of a review it is expected that all questions can be answered "yes". "No" responses indicate areas to be addressed by the appropriate members of the development team with a documented resolution prior to starting the next activity.

1.3.3 System Development Product Templates

Templates for each of the work products specified in the development lifecycle are provided in this section. They are intended to assist the development team in the creation of the products, but are not intended to limit how the information is created or presented. Document as the term is used here merely indicates a formatted representation of the work product. The templates provide guidelines on what should be included in the chosen representation.

Several documents identify subsections that could potentially exist as documents themselves. If the complexity of a project suggests that this should be the case, then Project Managers should exercise judgment whether to do so. Depending on project size, complexity, and risk, any of these documents may be written informally. Using the template as a guideline for the information presented is encouraged in these situations.

Much of the information specified in the templates will be subject to methodologies used on the project. Diagramming of software components, user interfaces, or any other development product may be adequately documented with the tool of choice. It may also be the case that material is presented, reviewed, and managed in electronic form rather than printed materials. When convenient and appropriate to do so, the use of digital representation is encouraged.

1.4 Related Documentation

The *NARA SDLC Guidelines* is a companion document containing individual detailed references of certain engineering or management disciplines that span the entire systems development lifecycle (e.g., Software Engineering, Configuration Management, Data Modeling, IT Security, Records Management, etc.). The references contained in that document are noted frequently throughout the lifecycle process described herein.

1.5 Common Terms

Some key terms are used throughout this document. The reader should have a clear understanding as to how they are applied here.

Product - The resultant system as described by the Product Plan.

Phase - One cycle of a development effort intended to produce a defined outcome. Phase types are defined under the MED systems model as testbeds, prototypes, pilots, initial operations capability, and evolutionary releases.

Project - A temporary effort to create a known set of deliverables

Activity - A discrete portion of a build phase intended to produce certain work products with a milestone review.

Task - A unit of work performed under an Activity. Usually intended to produce a defined outcome or to create input into another task.

Configuration Item (CI) -

An item or set of items managed as a single entity. Changes to a CI are maintained through a single release specification.

Requirements Traceability Matrix (RTM) -

A mechanism used to trace system requirements from their highest level to a built item primarily for the purpose of identifying the impacts of changes or to verify that all requirements are addressed. Can be created manually (e.g., with a spreadsheet) or through Requirements Management tools.

2 The Managed Evolutionary Development Process Model

2.1 Background

The Managed Evolutionary Development (MED) model described in this section is the preferred approach for all IT systems development within NARA. The MED model is similar to other IT development process models used throughout industry, but it uses a best-practices approach tailored to NARA needs. The following discussion identifies other models and how the MED model was developed for use within NARA.

2.1.1 Generic IT Development Process Models

There are three well-known generic IT development process models: Waterfall, Spiral, and Evolutionary. The development of these models has been partly driven by the need for better estimation and planning needs in software systems development. This, in fact, is one of the primary motivations for the implementation of a defined process for NARA IT systems. The lack of a process, sometimes referred to as the "Ad-hoc process," is characterized by unpredictable performance and an inordinate reliance on the accomplishments of individuals in the success of projects. The MED model develops a repeatable process that can be improved over time.

The MED model most closely resembles the Evolutionary model, but incorporates aspects of the others. The generic models are examined below with an indication of what elements are used in the MED and what elements are eliminated.

2.1.1.1 The Waterfall Model

The Waterfall Model is usually understood as the first, broadly used structured development process in Software Systems development. It is characterized by a set of formal activities that require sign-off by system owners before proceeding to the next activity. It is often derided for its reliance on heavy documentation and problems related to trying to get all requirements in a single massive effort. To be fair, the Waterfall was created in an environment when program development and changes were difficult and time consuming. Although application of the "classic" Waterfall is limited, it still provides the foundation for all other widely used development models in the software industry.

Elements included in MED from the Waterfall Model are a base set of activities, work products, and milestone reviews. The MED model seeks to minimize problems inherent in the Waterfall including late risk management, difficulty in changing requirements, and long waits before users are provided any system functionality.

2.1.1.2 The Spiral Model

The Spiral Model is a more modern approach to software systems development that seeks to address some of the pitfalls with the Waterfall particularly in the areas of risk

management and late system implementation. Advances in tools that ease program coding and creation of visual models have also contributed to its adoption. These advances led to the use of prototyping (mostly in the area of user interfaces) that enabled potential users of the system to get a better feel for how the system would actually operate.

The Spiral model seeks to start small and work outwardly to create a defined operational system. By evolving prototypes through the development of an operational system, the model allows users to be involved in assessing the system. Risk management is embedded in the process by having milestone reviews and system evaluations conducted at the end of each process cycle (or "spiral").

A recent version of the Spiral Model adapted to Object-Oriented development (and predominantly the Unified Modeling Language approach) refers to the cycles of Inception, Elaboration, Construction, and Transition. Activities in this model continue to map to the foundation activities developed under the Waterfall, but the terminology can lead to a perception that it is a revolutionary model. It is important to note that within NARA systems development that the MED model is flexible enough to accommodate the OO paradigm. This will be discussed more within the details of the MED approach.

Elements of the Spiral model that have been incorporated in the MED include early risk management, prototyping, and user involvement. The MED model seeks to eliminate the use of prototype design and code in the operational environment.

2.1.1.3 The Evolutionary Model

The Evolutionary Model is also iterative in nature, but differs from the Spiral Model by seeking to produce operational systems from each phase and early in the process. System needs are decomposed into phases of limited scope that can be easily implemented and then evolved with incremental releases that expand functionality. It may also use elements such as prototyping, but only to derive requirements or assess alternatives. The MED model adopts the approach of developing a defined product from each phase (although not necessarily an operational one).

The Evolutionary Model does have some pitfalls similar in nature to the Waterfall. Namely, requirements collection and design efforts necessary to implement an operational system can bog down the process. The MED avoids this through the use of testbed, prototype, and pilot phases.

2.1.2 Adaptation of the Generic Models to MED

As noted above, the MED model uses practices and techniques from each of the generic IT development process models to formulate a comprehensive set of processes. Other drivers in the creation of the MED model include factors such as a push towards COTS based systems, Object-Oriented methodologies, and advances in CASE and visual development tools.

2.1.2.1 COTS Based Systems

Most systems within NARA are expected to use a COTS based approach to address their needs. This means minimal custom development associated with the system development. When evaluating how certain tasks apply to a project, the impact of the COTS items must be evaluated to ensure development tasks are not duplicated.

Another issue with COTS based systems is that it requires more design and market analysis to occur up front during concept and requirements definition activities. This is particularly true when the available packages must be implemented "as-is" and may influence system requirements (i.e., business processes). The NARA model adopts an approach that encourages development of the system architecture and design elements to occur early in the process, partially through the use of prototype and pilot phases.

2.1.2.2 Object-Oriented Methodologies

The activities described under the MED model are to be applied to all projects regardless of the development methodology being used. An OO approach should not eliminate the milestone reviews or work products described herein. Rather, the OO methodology being used should only impact how the information is presented.

An effort has been made to include OO concepts and terminology in the process descriptions to show where and how an OO methodology can be applied.

2.1.2.3 New technologies

Newer tools and techniques often have an effect of compressing the system development activities. It is accepted and expected that new technologies facilitate the activities by, for example, having a design and program code created with CASE or visual development tools. In this case, the life cycle is compressed because design actually occurs through abstraction in the development tools. New technologies do not, however, replace the necessity of understanding the system that needs to be built, ensuring system components work together correctly, or testing of the system to ensure proper operation.

The NARA model is flexible enough to accommodate compression of activities. It continues to emphasize, however, that tasks such as requirements analysis and testing are performed regardless of how it actually takes place.

2.1.3 Special Considerations in application of the MED lifecycle

Predetermined Solution. The activities outlined in this document are intended to lead to the best solution to a problem. Occasionally, a particular technology solution is deemed to be the predetermined solution, determined before the problem has been fully defined or understood. This may be because a package already exists in-house with expertise readily available, because there are desirable features not offered by other products, etc.

When faced with predetermined solutions the activities are still performed although the focus may shift from solution definition to risk mitigation. For example, the development team may need to ensure that the system environment will support all the required activities and will need to spend additional time performing system integration tasks. Issues that may impact the success of the project must be identified and addressed at the earliest opportunity.

2.2 Managed Evolutionary Development (MED) Model Overview

The MED model is composed of *phases*, which align with system *capabilities*. A phase is one pass through the lifecycle activities that culminate into a system product such as a testbed, prototype, pilot, or operational capability. The MED Model leads to a series of increasing system capabilities or pre-planned product improvements. A system capability may expand by any of a number of dimensions including an increase in functional capability, increase in network and/or storage capacity, or an increase in user access.

Development *strategies* are employed to progressively move towards an operational capability. For example, a development strategy may include building a prototype in Phase 1 to demonstrate a proof-of-concept for the use of a specific technology. Phase 2 may develop a pilot system to expand user access to the technology and Phase 3 may develop the first version of the system to be approved for production use and be based upon reuse of requirements and architectural components from Phases 1 and 2.

The following sections include a description of the MED phases, system development strategies, foundation activities, reviews, and products.

2.2.1 MED Phases

The MED system development model typically includes two or more lifecycle phases before producing the "initial operational capability" (IOC). The total number of phases is estimated in the Product Plan, but may be unknown until after execution of one or more phases. Subsequent revisions of the Product Plan can modify phases based upon experience. This is because the MED model assumes an evolutionary “learn as you go” methodology. For systems that are too complex to understand up front, small, manageable pieces of known issues or risks are resolved before moving to the next set of issues/risks.

After completion of each phase an evaluation is made to determine whether phase completion objectives have been met. The following sections describe each phase in detail.

2.2.1.1 Prototype

A prototype is the infant stage of system development. It is frequently the first step in developing an initial concept of operations. It provides proof-of-concept at a level that end-users and senior managers are able to visualize. Prototypes are generally developed to demonstrate technology capabilities to end-users and managers, to define and refine business process requirements, and/or to evaluate a part of the system technology. Since prototypes are learning tools and are experimental, not all components may be reusable, will not necessarily be scalable, will have limited scope, and will have limited, if any, interfaces with legacy IT systems. Prototypes should not be incorporated into an operational system, although concepts, requirements, and hardware and software may be reused in subsequent phases of development.

2.2.1.2 Pilot

A pilot is the adolescent stage of system development. Surpassing the capability of a prototype, it usually includes scalable technology, additional functionality, end-user access, and/or incorporation of real interfaces with other IT systems. A pilot is generally used to test an initial operational capability in an environment that closely resembles the desired end product and involves at least a subset of the end-users. Depending upon its perceived success at the end of the pilot evaluation stage, a pilot may be deemed an initial operational capability with a limited user base. Still, the possibility exists that the pilot may not perform as desired and improvements are required before meeting operational objectives.

2.2.1.3 Testbed

A testbed is a Research and Development (R&D) platform or laboratory for testing and evaluating technologies. It is envisioned that testbeds will be used within NARA to research the use of new technologies and to experiment with combinations of new and existing technologies. A testbed is the embryonic stage of system development. It provides a technology green-house for conceiving ideas for potential implementation. A testbed can be used as a test environment throughout the life of a project. They can be used to test and evaluate commercial-off-the-shelf products as potential system solutions. Testbeds can be devised to provide mock-up, simulated, or live interface testing for integrating multiple components of a system or multiple systems.

Though defined as a phase under the MED Model, testbeds may actually be implemented throughout the system development effort. The intent of referring to testbeds as a phase is to set the expectation that testbeds are created with a defined objective and the standard process is followed for implementation and evaluation of its success.

2.2.1.4 Production (or Operational) System

The primary strategic goal for every project is to eventually achieve a capability for a new IT system that is approved for production use. A system is ready for production use

only when it meets business objectives and successfully implements business processes, utilizes reliable technology; meets performance, maintainability, availability, security, and other system requirements; meets reach, range, and maneuverability objectives; and adheres to the IT Standards and Guidelines. Other phases are used to evolve system capabilities into a production system.

Though defined as a phase under the MED Model, testbeds may actually be implemented throughout the system development effort. The intent of referring to testbeds as a phase is to set the expectation that testbeds are created with a defined objective and the standard process is followed for implementation and evaluation of its success.

Table 2-1 summarizes each phase.

Table 2-1 Phase Summary Description

Prototype
• "Throw-away," with some reusable components • Used to refine requirements • Has limited scope • Demonstrates a "proof-of-concept" • Has limited interfaces with other IT systems (e.g. mock or simulated) • Technology not necessarily scalable • Can be "fed" by a Testbed • Must be evaluated before escalation to pilot or production capability
Pilot
• Real interfaces with other IT systems • Provides an interim capability • Technology should be scalable • Can be fed by Testbed and/or prototype • "Means" not an "end" • Demonstrates ability to meet business needs • In an evolutionary sense, a pilot can be approved for production use at the end of the pilot evaluation period (Note: All other infrastructure components must be in place) • Must undergo systems engineering (minimal Waterfall activities) before the pilot can be declared to be approved for production use
Testbed
• Research and Development (R&D) platform/laboratory • Technology green-house • Experimental • No operational impact • Potentially broad scope • Can be used throughout lifecycle to test hardware/software changes • Can be used to replicate production environment without affecting or disrupting the production environment • Can feed prototype, pilot, or production system
Production System
• Tested/evaluated/proven • Supported by infrastructure (and system engineering methodology) • Scaled to dimension of problem • Meets reach, range, and maneuverability objectives • Meets business objectives for the phase subject to continuing improvement and technology refresh • To receive approval for production use, must at least incorporate the Waterfall activities associated with Preliminary Design through Acceptance and Transition (assuming that the Concept Development and Requirements Definition activities have been completed as pre-cursor (pre-development effort) activities)

2.2.2 System Development Strategies within the MED Lifecycle Model

Product development employs a strategy designed to minimize risk and achieve an operational capability. The strategy for a product uses one or more development phases as defined earlier to evolve the system from concept to operations.

Each development plan will use a combination of development phases following the general outlines described here. A general order of system maturity places the strategies in the maturity scale depicted in Figure 2-1. As illustrated, a testbed can be used anywhere within the entire product lifecycle.

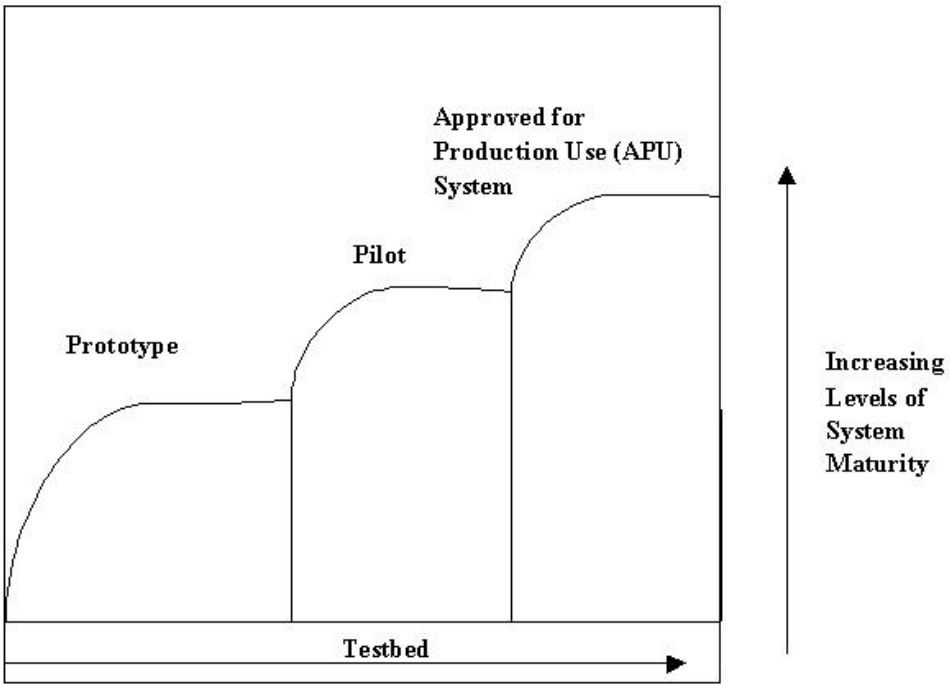

Figure 2-1 Maturity Scale

It is important to note that while a pilot is a more mature state of development than a prototype, a pilot does not always need to be developed before the operational system. Similarly, a prototype may not be needed. The "plug and play" usage of the strategies depends upon successful mitigation of the identified and prioritized risks and achievement of completion objectives defined for each project or phase.

2.2.3 Guidelines for Selecting Strategies

Development strategy selection for the most part is a creative endeavor matching an infinite variety of solutions against a limitless set of problems, conditions, or risks. Therefore, it is impossible to capture all of the potential solutions to a given problem. However, Table 2-2 provides some recommendations for selecting a strategy when certain risks or conditions are present.

Table 2-2 Strategy Recommendation

Problem/Condition/Risk	Possible Strategy
Functional and data requirements are not well-defined or are in flux.	Design a prototyping and/or piloting effort to help end-users validate or refine the functional, data, and user interface requirements.
Technology is not mature; it has not been successfully demonstrated in industry for at least 4-6 months.	If the technology is brand new consider waiting for technology to mature within industry. Follow an IT policy of leading edge; not bleeding edge. Consider establishing a testbed as a first MED phase or propose a separate Product Plan merely to do precursor activities. If the development effort cannot wait for technology to mature in industry, consider evaluating a number of technologies against requirements and predefined criteria. Select a mature technology that meets requirements and will be maintained by the vendor in the long-term. Highlight this risk in the Product Plan. Select a testbed to evaluate COTS alternatives for MED Phase 1. Reevaluate the Product Plan after Phase 1 of the MED. If the technology has been used in industry for a short period of time, it may be best to demonstrate its ability to perform in the specific environment before committing to its full-scale use. Its capability can be evaluated in a testbed environment.

Problem/Condition/Risk	Possible Strategy
The infrastructure architecture must be upgraded to support the new system.	Pilot the system in an upgraded, infrastructure mock-up testbed environment. Recognize the organization risk. Propose a MED strategy with a phase for a pilot.
System requirements for 3-tier architecture such as performance, reliability, availability, maintainability, security, usability, reach, range, and maneuverability requirements are not well understood.	Use prototyping to test interoperability aspects (reach, range, and maneuverability) between system components and external interfaces. Propose a MED strategy with a prototyping phase. Perform stress testing for security, scalability, or other system requirements in a testbed environment. Propose to establish a testbed as the first MED phase and maintain the testbed throughout the development lifecycle. Use piloting efforts to evaluate technical performance in conjunction with user satisfaction. Propose a MED phase with a pilot, in addition to a prototype and testbed.

2.2.4 Activities in Managed Evolutionary Development (MED) Build Phases

Each project phase requires that one or more "Foundation Activities" be performed. The NH-tailored model consists of the following activities:

- Concept Development
- Requirements Definition
- Preliminary Design
- Detailed Design
- Development
- Integration and System Test
- Deployment and Acceptance
- Production
- Retirement and Rollover

A sequential pass through these activities constitutes one development phase. Associated with each activity is a set of products that may include a combination of documentation, software, and hardware components. Also associated with each activity

is a milestone review, which obtains consensus among the Project Team members and ensures the project is progressing according to plan.

The diagram below shows the Activities and coverage by phase. A brief discussion of the various phases and the expected activity effort follows.

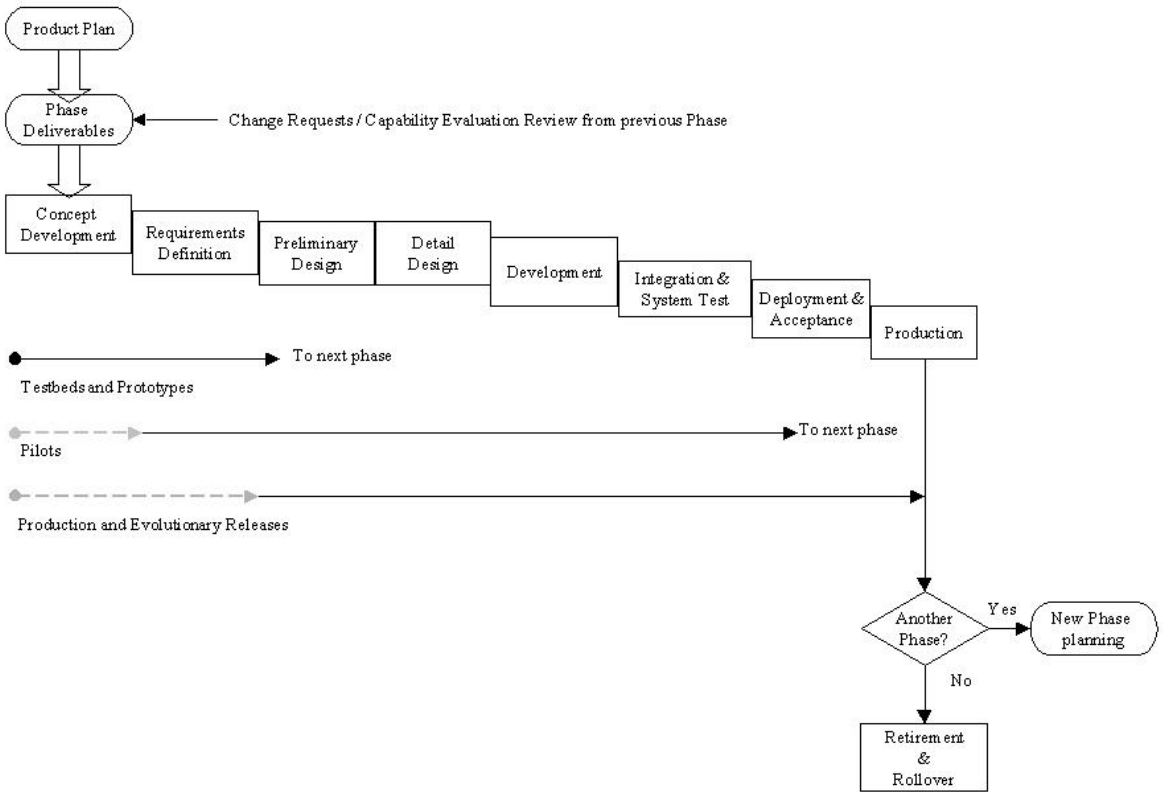

Figure 2-2 MED Activities by Project Phase Overview

Testbeds and Prototypes

Testbeds and Prototype phases are expected to cover Concept Development and Requirements Definition activities at a minimum. When used, they will typically be evaluated for lessons learned and to feed a Pilot or Production phase. It should be noted that large systems testbeds and prototypes often cover only a single system component and may not eliminate the need to perform Concept Development or Requirements Definition in later stages of the project.

Pilots

Pilot phases are required to execute the Requirements Definition activity through Deployment & Acceptance. The Requirements Definition activity is necessary to formally capture and review specifications even if a prototype phase was conducted previously to identify them.

If the Prototype Phase was not performed for the project, the Concept Development activity must be included as part of the Pilot Phase.

Initial Operations Capability (IOC) / Evolutionary Operational Releases

IOC and Evolutionary releases require all activities from Preliminary Design through Production.

Requirements Definition is required if not defined in a previous phase. Requirements tasks are also required when new system components are being added or when system scale is being significantly changed (e.g., from departmental pilot to full Agency production).

Concept Development is required if not performed in previous phases or new system components are being added.

Retirement and Rollover is performed only when the system is retired or replaced and is not associated with a particular build phase.

2.2.5 MED Milestone Reviews

During a MED phase, multiple reviews may take place. For example, if the phase comprises a prototype development effort a Concept Review and Requirements Review may be required for the development of the prototype itself. Since the main objective of the prototype will be to resolve issues leading to an Operational Capability, a review must take place at the end of the prototyping effort to determine whether the specific capability met its completion objectives (as depicted in the Project Plan). Moreover, the

reviews assess whether moving to the next planned step is viable or a revised Project Plan is necessary.

The MED methodology always includes one review at the end of each phase. This is the Capability Evaluation Review (CER) wherein completion objectives are assessed and a go/no-go decision is made for the next planned phase. While the CER is not included as part of the activities, it is required to capture lessons learned and as a risk mitigation technique.

2.2.6 MED Product Requirements

As in the case of the MED Reviews, the MED product requirements for a single phase generally correspond with the activities performed. For example, in the case of a requirements definition activity, a Requirements Document would be a required product. If a MED phase included a full-scale operational activity, the complete set of products would be required. Work products may be simplified for the development of prototypes or pilots since the main objective of each is to obtain more information to develop an operational capability. Decision regarding effort on work products are made during the Product and Project planning stages.

2.3 Activities, Reviews, and Products

This section briefly describes the purpose and objectives of each of the foundation activities in greater detail. The expected milestone reviews and work products are described with each activity. Sections 3 through 11 of this document provide the detail tasks, review checklists, and work product templates for each activity defined here.

The following diagram graphically represents each of the major activities, reviews, and products that may occur during one development phase.

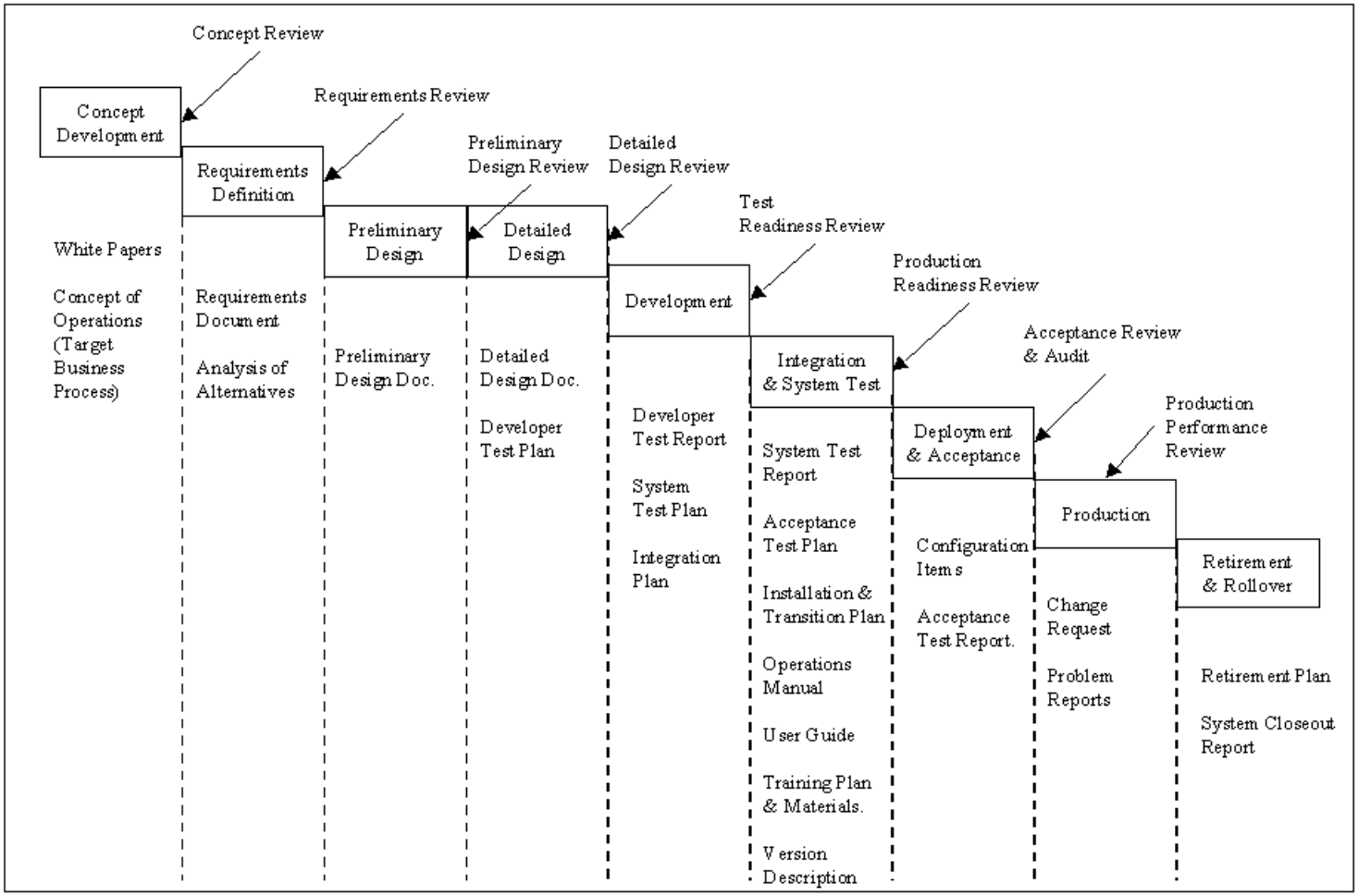

Figure 2-3 Activities

2.3.1 Concept Development

2.3.1.1 Concept Development Activity

The purpose of the Concept Development activity is to identify system alternatives capable of satisfying the mission need. This activity consists of research and comparison analysis and usually involves high-level evaluation of existing Commercial-off-the-shelf (COTS) products. It may also encompass a Business Process Reengineering effort.

Prototyping and other analytical methods are used to better define the scope, boundaries, and feasibility of the system. While alternatives are analyzed during this period, it is important to note that a firm decision on an approach is not made until after the requirements activity.

2.3.1.2 Concept Review (CR)

The review that takes place at the end of the Concept Development activity is the Concept Review (CR). The purpose of the Concept Review is to ensure that the "system-of-systems" level or cross-project interface requirements and the top level system requirements are understood by the Project Team before proceeding with a specific project. If the Concept Definition activity includes a Business Process Reengineering (BPR) effort, the Target Business Process (i.e., the proposed new business process) must be presented and evaluated in conjunction with the high-level system architecture.

2.3.1.3 Concept Development Products

The products generated during the Concept Development activity include the following:

- White Papers
- Concept of Operations Document (Target Business Process)

These documents are described in the following sections. During Concept Definition, other activities such as alternative analysis, business process reengineering, etc. may require additional products not yet identified.

Concept of Operations Document

The main document produced during the Concept Development activity is the Concept of Operations (ConOps) Document. The ConOps describes a proposed system in terms of the user needs it will fulfill, its relationship to existing systems or procedures, and the ways it will be used.

If a Business Process Reengineering effort is performed during the Concept Definition Phase, the target business processes that the system will automate need to be included in the ConOps, along with the business process interfaces with external systems and high-level manual procedures.

White Papers

White Papers are optional products that can be developed at any point within the lifecycle, but most often are written during the Concept Development activity. White Papers are informal descriptions of ideas. It is common to develop one or more white papers to initiate discussion between users, managers, and technical personnel on proposed automated solutions to address business needs. Once ideas turn into a more solid proposal, a Concept of Operations is developed to document the consensus.

2.3.2 Requirements Definition

2.3.2.1 Requirements Definition Activity

The purpose of the Requirements Definition activity is to identify requirements for the proposed system including processing, data, and records management requirements; interfaces between other systems; user interfaces; and miscellaneous needs such as reliability, availability, maintainability, performance, security, reach, range, maneuverability, personnel, and training requirements.

2.3.2.2 Requirements Review (RR)

The purpose of the Requirements Review is to establish a mutual understanding among the Project Team members regarding the documented requirements. The requirements are reviewed from both needs in the current phase as well as allowing room for growth in future phases.

2.3.2.3 Products

The products generated during the Requirements Definition activity include the following:

- Analysis of Alternatives
- Requirements Document

Requirements Document

The requirements document specifies the requirements for a system or subsystem, its software components, and its external interface and communication requirements between systems, subsystems, hardware components, software components, or manual operations.

Analysis of Alternatives

This effort establishes a draft of the recommended system architecture and an evaluation of alternative approaches. Alternatives analysis may consist of identification and evaluation of supporting technologies (including COTS products), “make or buy” pros and cons, trade-off analyses of functional and data allocation to different system configurations, impact assessments on the NARA IT architecture, security alternatives, and a cost benefit analysis of each option.

2.3.3 Preliminary Design

2.3.3.1 Preliminary Design Activity

The purpose of the Preliminary Design activity is to define the high-level hardware configuration and logical software and database design for each of the components and software applications that constitute the system. All COTS products, reusable components from other sources, and upgrades to legacy systems are described where applicable. Rationale for the design selection is described in terms of technical, cost, and schedule benefits.

2.3.3.2 Preliminary Design Review

The purpose of the Preliminary Design Review is to gain mutual understanding among the Project Team members of the logical design of the system components including hardware configuration, software, personnel, and data design and interaction.

2.3.3.3 Products

The products generated during the Preliminary Design activity include the following:

- Preliminary System Design Document

Preliminary System Design Document

A Preliminary System Design Document portrays high-level design of the system components including the Implementation Data Model for database items, hardware component configuration (e.g., client-server configuration), allocation of software components to hardware components (e.g., distributed or centralized databases), network services and communication protocols between platforms and external interfaces. Other details include computer hardware processing speed, memory, and hard-drive capacities, software design, software interface design, middleware configuration and interface standards and design, and security design.

2.3.4 Detailed Design

2.3.4.1 Detailed Design Activity

The purpose of the Detailed Design activity is to describe the physical design of the hardware configuration, software, and database components of the system. The design documentation should provide enough information for developers to construct and implement the system.

2.3.4.2 Critical Design Review

The purpose of the Critical Design Review is to gain mutual understanding of the physical design of the system components, including interface descriptions, communication protocols, facility and network layout, physical database design, and programming specifications.

2.3.4.3 Products

The products generated during the Preliminary Design activity include the following:

- Detailed System Design Document
- Developer Test Plan

Detailed System Design Document

The Detailed System Design Document describes the physical design of the system to the lowest level components. It includes the physical data model (depicting the tables, fields, and primary keys), the detailed interfaces (network services, communication protocols, APIs, etc.), the software logic specifications, and interface/COTS products' parameter settings.

Developer Test Plan

The Developer Test Plan describes the plan for testing all individual components of the system and integration of the system components. It describes the tests to be performed, the environment to be used for testing, and provides schedules for test activities.

2.3.5 Development

2.3.5.1 Development Activity

The purpose of the Development is to construct and test components specified in the design. The System Test and Integration plans are also developed during this activity.

2.3.5.2 Test Readiness Review

The Test Readiness Review is conducted after the developer has performed unit and component testing (i.e. "developer testing"). The purpose of the Test Readiness Review is to ensure that the team is ready to begin formal system testing. The adequacy of the System Test Plan and Integration Plan are also examined.

2.3.5.3 Products

The products generated during the Development and Integration activity include the following:

- Developer Test Report
- System Test Plan
- Integration Plan

Developer Test Report

This report summarizes the developer testing performed against constructed items. Of primary importance here is an assessment of the testing effectiveness and areas that encountered significant difficulties.

Integration Plan

The Integration Plan describes how the individual system items are to be delivered and integrated to construct the defined Configuration Items. For custom software this will include such items as how the software must be compiled and installed on to the target platform.

System Test Plan

The System Test Plan describes the test preparations, test environment, test cases, and test procedures to be used during full-scale system level testing. At this level the testing will focus on the ability of the system to meet the requirements and interact with external systems.

2.3.6 Integration and System Test

2.3.6.1 Integration and System Test (I&T) Activity

The I&T activity begins by integrating system components into defined Configuration Items. System testing is conducted to validate the built system against the requirements. This activity also prepares for the transition of the development product into a production environment by addressing data migration, training, maintenance and other deployment issues.

2.3.6.2 Production Readiness Review

The purpose of the Production Readiness Review is to determine if the system is complete, tested, and ready to deploy into the production or operational environment. The adequacy of maintenance and support readiness as portrayed in the Users Manual, Operations Manual, Training Materials, Installation & Transition Plan, and the Version Description Document is also determined.

2.3.6.3 Products

The products generated during the Integration and System Test activity include the following:

- System Test Report
- Acceptance Test Plan
- Installation and Transition Plan
- Training Plan
- Training Material
- Operations Manual
- Users Manual
- Version Description

- Records Schedule ready for NARA approval process

System Test Report

The System Test Report is a record of test results of each system component and the integration of the components. It provides a summary of results and specifies any remaining problems.

Acceptance Test Plan

The Acceptance Test Plan is the testing plan specified by the Project Manager and includes acceptance criteria, method for testing (observation of developer testing or independent testing), test environment and preparation, tests to be performed, test data to be used, and schedules for test activities.

Installation and Transition Plan

The Installation and Transition Plan is the plan for installing and transitioning system components from the development environment to the user sites, and includes considerations for user training, conversion from existing systems, and replacement-system retirement. It identifies the hardware, software, and other resources needed for lifecycle support and describes the tasks and schedules for transitioning deliverable items to the support agency.

Training Plan

The Training Plan includes training activities and schedules for system end-user and system support staff training for components of the system. It should additionally specify how users and support personnel have a certifiable understanding of the system operations.

Training Material

Training Material includes training curriculum, courseware, job performance aides, online help packages, or other material concerning the use, operation, or maintenance of the system components.

Operations Manual

The Operations Manual provides information needed to operate a given computer system and its peripheral equipment. It also provides system administrators information on how to install and operate a software component in a centralized or networked environment.

Users Manual

The Users Manual provides a hands-on user with information on how to operate the system. Elements of the Users Manual usually include an overview of commonly performed processes and troubleshooting tips (along with any help desk information).

Version Description

The Version Description provides an "as-built" depiction of the system hardware configuration and contains or references the executable software, source files, and hardware and software support information, including software compilation, build, and modification procedures, and hardware and communications network parameter settings.

2.3.7 Deployment and Acceptance

2.3.7.1 Deployment and Acceptance Activity

The purpose of the Deployment and Acceptance activity is to install the system in the operational environment (possibly in a limited capacity) and conduct acceptance testing to ensure that the system and its associated products perform in accordance with specified technical and contractual requirements. Maintenance, training, and data migration activities are completed and tested. This is a formal transition between the developers and the acquirers/end-users of the system.

2.3.7.2 Acceptance Review and Audit

The purpose of the Acceptance Review and Audit is for the acquirer, technical support, and or end-user to formally evaluate and accept a deployed and fully tested system from the developer. Acceptance may consist of acceptance testing by a developer-independent party, observation of formal testing by the developer, and/or a functional and physical configuration audit which compares the development product against the functional, data, and system requirements, design specifications, and contractual requirements.

2.3.7.3 Products

The products generated during the Deployment and Acceptance activity include the following:

- Configuration Items
- Acceptance Test Report

Configuration Items

The built or customized components of the system consisting of the executable software, source and object code, software libraries, software licenses, hardware, datasets, interfaces (e.g. APIs), and network environment, and software, hardware, and network maintenance agreements, and all updated system documentation. An inventory list should accompany the products.

Acceptance Test Report

The Acceptance Test Report includes the test results from the acceptance tests and includes a final recommendation for acceptance.

2.3.8 Production

2.3.8.1 Production Activity

Operations and maintenance of the system, or in-service engineering, occurs during this activity. In-service engineering includes bug fixes, development of scripts, or emergency workarounds to ensure the operability of the system. If it is determined that the system needs major enhancements (reengineering or replacement), a new project must be specified through development of a Product Plan.

2.3.8.2 Production Performance Review

The purpose of the Production Performance Review is for the Product Owner, NARA end-users, and NH operations/project staff to periodically discuss and assess the state of the operations of the production system, identify problems, discuss potential solutions, and plan implementation of simple fixes.

Also, this review may provide a forum to discuss the need for major enhancements and the development of a Product Plan to propose a new system version. Similarly, it may be proposed that the system be retired.

Eventually, NARA may establish this review in the form of an on-going Configuration Control Board to manage changes to the software, hardware configuration, and the associated documentation of a production system.

2.3.8.3 Products

Production products include hardware and/or software updated through bug fixes or additional scripts, and updated documentation which notes the fixes. A change control process, to include change requests, should record problems, recommended solutions, and actual changes to the system. In general, products include:

- Change Requests
- Problem Reports
- Operation Status Reports
- Revised Records Schedule (if required)

2.3.9 Retirement and Rollover

2.3.9.1 Activity

This activity addresses the retirement and possible rollover of a production system. This activity is invoked when the Product Owner and NH formally decide to retire, replace, or redesign the existing production system. The decision may include terminating the system on a specific date, phasing out the system gradually, or operating the system in parallel while a new or replacement system is being designed and developed. During the interim phase, the system will continue to

operate in production mode under a scaled down version of the production operation activities identified in previous section. However, this activity begins the formal process of planning for the retirement and possible rollover of the system.

2.3.9.2 Review

The Retirement/Rollover Review occurs at the end of the Retirement/Rollover activity for the purpose of performing final review of all retirement and rollover activities. All retirement and rollover plans and reports are reviewed. Also, all records management plans and schedules associated with the product, project, and system are reviewed and discussed.

2.3.9.3 Products

The Retirement and Rollover products include:

- System Retirement and Rollover Plan
- Data Conversion Plan (if appropriate)
- System Close Out Report
- Product/Project/System Documentation and Files
- Product/Project/System Case Files (if appropriate)

The next sections examine the detail of each of the defined activities, reviews, and work products.

Detail Activity Information

(This page intentionally left blank)

3 Concept Development

3.1 Purpose

Concept Development tasks are initiated at the beginning of a development phase so that the development team can better understand the user's needs, assess alternative approaches, and begin to outline a solution for the desired system. Either Government or Contractor personnel may perform Concept Development.

While this activity is somewhat abstract, the goals for the activity is to develop top level system requirements and a solution that

- is cost effective (though not necessarily the most technically elegant);
- provides clear benefits for users of the system;
- is achievable with resources allocated to the project; and
- gains Product Owner commitment to the approach.

3.2 Concept Development Task Outline

The following timeline figure shows the typical sequence of Concept Development tasks and work product creation.

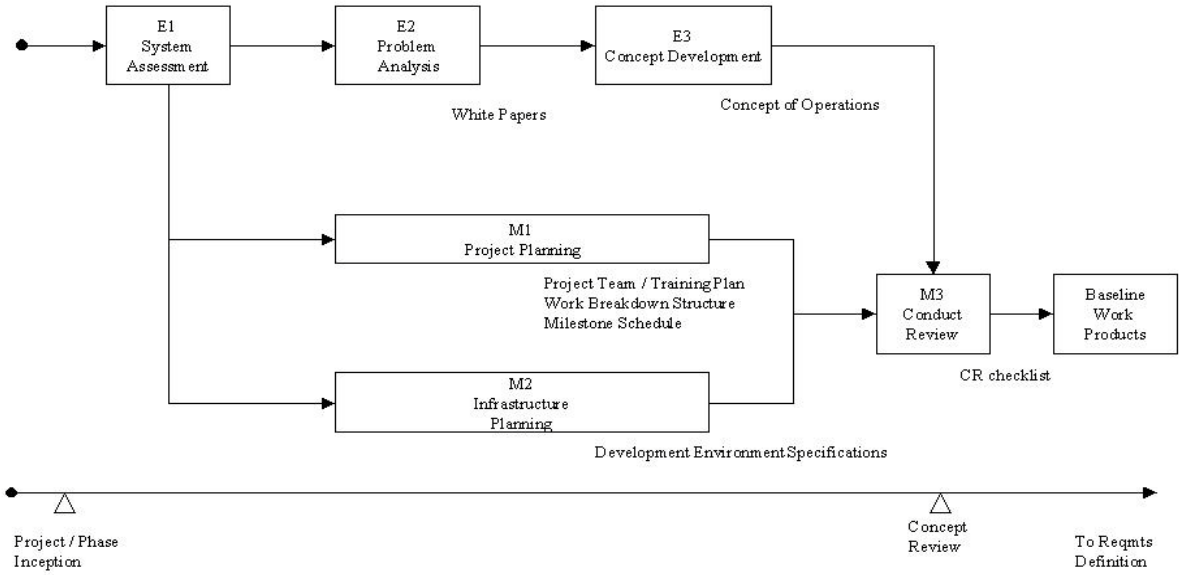

Figure 3-1 Concept Development Activity Timeline

Table 3-1 Concept Development Performance Requirements

Entry Criteria	**Concept Development Entry Criteria**
Products and activities to be completed before beginning a new software development activity.	Product Plan Draft Project Plan - Key personnel defined - Project risk assessment (i.e., Preliminary Risk Assessment and Security Categorization) performed - Phase deliverables defined Project Kickoff held
Work Products	**Concept Development Work Products**
Products and activities to be completed during the current software development activity.	Solution White Papers Concept of Operations to include - 1st level System Engineering Diagram - Conceptual Data Model - Business ConOps aligns with the Business Architecture - High-level system architecture concept is expressed within the CONOPS - System architecture concept aligns with the Enterprise Architecture (EA) system partitioning, EA principles, and EA guidance - System concept conforms to NARA standards (See NARA EA, Appendix B – Technology Standards Profile) - Security cost considerations and reporting Refined Project Plan to include - Work Breakdown Structure (WBS) and Schedule - Project Team definition (Roles / Responsibilities) - Staffing & Training Plan - 801 EA Compliance criteria have been satisfied Completed CR Checklist

Exit Criteria	Concept Development Exit Criteria
What must be finished before an ongoing activity is designated complete	A documented resolution for each issues identified during a Concept Review Detailed schedule for Requirements Definition activity Baseline ConOps Product owner accepts ConOps Guidance team approves new Project Plan elements and continuation to next activity

Engineering Tasks

E1. Assess user expectations and measures of effectiveness
   a. Conduct user interviews and evaluate existing systems to attain understanding of the new system need
   b. Define and document top level requirements
   c. Define the evaluation criteria to be used for proposed solutions
   d. Identify system constraints (cost, time, etc.)

E2. Analyze Problem
   a. Resolve conflicting requirements
   b. Define procedures for analyzing proposed solutions, which may include
      i. Structured Analysis / Business Process Analysis / Use-case definition
      ii. Trade studies / Cost-Benefit analysis
      iii. Comparison of similar systems
      iv. Proof-of-concept prototyping
      v. Subject Matter Expert input
   c. Identify solution alternatives
   d. Analyze the competing candidate solutions against the established evaluation criteria.
   e. Identify the best-fit solution that satisfies the established evaluation criteria.

E3. Develop the Concept of Operations by describing the interaction of the system, the user, and the environment, that satisfies the operational need.
   a. Identify major interactions among users and systems.
   b. Use products provided by Business Process Reengineering (BPR), if applicable, to confirm interactions between users and systems. BPR may be performed prior to or as part of the Concept of Operations activity.
   c. Define logistical and operational support needs using products provided by BPR where applicable.
   d. Develop technical cost estimate of proposed solution by component with justifications.

Management Tasks

M1. Develop new project plan elements
   a. Define project roles required for implementing the recommended solution
   b. Create a baseline WBS for recommended solution
   c. Develop project staffing / resource allocation plan
   d. Prioritize and schedule milestones for development
   e. Define development staff training requirements and plan

M2. Plan Development Infrastructure (facilities, development tools, etc.)
   a. Identify equipment and tools needed for construction of proposed system
   b. Identify facilities for staff and equipment
   c. Prepare acquisition of infrastructure according to NARA procedures

M3. Concept Review (CR)
   a. Prepare and Distribute CR materials (checklist, work products)
   b. Review materials prior to CR meeting
   c. Conduct CR meeting
   d. Resolve open issues
   e. Establish configuration baselines for work products

Roles	Responsibilities
Product Owner	• Provides clarification regarding product needs and objectives (E1) • Sets measures of effectiveness (E1) • Approves ConOps (M3)
Guidance Team	• Ensures adherence to NARA Enterprise Architecture and Information Technology Standards (M3) • Approves of new project plan elements (M3)
Project Management	• Ensures proposed solution concept fits within project resource constraints (E2) • Prepares (or oversees preparation of) new Project Plan elements (M1) • Plans facilities for development staff (M2) • Conducts Concept Review (M3) • Completes Concept Review checklist (M3)
Systems Engineering	• Conducts user interviews and gathers analysis data (E1) • Prepares white papers for system solution alternatives (E2) • Prepares system engineering diagrams (E3) • Prepares technical cost estimates (E2 / E3) • Prepares and presents Concept of Operations (E3 / M3) • Provides development facility requirements (M2)
Data Management	• Identifies external data interface requirements (E2) • Provides input to solution alternatives for data management layer(s) (E2) • Prepares Conceptual Data Model (E3)
Software Engineering	• Provides input to solution alternatives for the system's presentation and data processing layer(s) (E2) • Prepares white papers regarding software solutions (E2)
CM	• Captures work products and establishes baseline (M3)
QA	• Assesses and approves adequacy of work products prior to Concept Review (M3) • Ensures completion of CR checklist (M3)
Test and Evaluation	• Prepares Concept Review evaluation checklists (M3) • Evaluates ConOps in accordance with defined checklist (M3)
Operations and Maintenance	• Identifies logistical and operational needs for proposed system concept (E3)

3.3 Concept Review Checklist

Purpose and Objectives:

The Concept Review demonstrates a base level of understanding between the Product Owner and the development team as to what the end result of the development will look like and how it will be constructed. As the problem is still being dealt with at an abstract level, it is expected that this review will be mostly subjective. However, use of defined acceptance criteria for the Concept of Operations is intended to provide a level of objectivity to the process.

Objectives for the Concept Review include

- Establish agreement between Product Owner, Guidance Team, and development team on the end result of the project.
- Demonstrate the feasibility of the proposed solution.
- Demonstrate a basis for cost, project team, and schedule estimations.
- Provide a basis for continued efforts towards developing detailed system requirements

Presenters

- Project Management,
- Systems Engineering

Participants

- Product Owner(s)
- Guidance Team
- Appropriate development staff (for project familiarization)
- QA

When:

The CR is conducted after the key development staff have assessed the system and developed an operational scenario as defined for the Concept Development activity and after QA has confirmed adherence to standards.

Key Issues

- Has a technically feasible solution been defined for the system need?
- Do estimates for the cost and schedule have a sound justification?
- Can the proposed solution be done with the resources allocated to the project?
- Have the appropriate types of expertise been identified to perform the project?

Entry Criteria

The following items should be done prior to conducting the review

Distribute the Concept of Operations 1 week prior to review
Distribute the Project Plan items 1 week prior to review

Exit Criteria

The review process is not considered closed until the following items are complete.

- All issues raised during the review have a documented resolution
- The Product Owner agrees to and signs the Concept of Operations
- The Guidance Team approves and signs the revised Project Plan (thereby agreeing to project continuation)
- QA confirms completion of the Review Checklist

Presentation of Concept of Operations

General approach to development of ConOps
Assumptions used in preparation of ConOps
Introduction and overview of system need, concept evaluation criteria, and general system description
Justification for system and nature of changes to existing processes
Description of operational environment / major system components
Conceptual Process Model and Conceptual Data Model descriptions
Description of interactions of system components with external systems and user interfaces
Presentation of operational scenarios and support needs
Description of internal interactions among system components
Impacts of proposed system
Analysis and comparison to alternatives
Cost breakdown of proposed system
Open Q&A

Complete the following checklist for the Concept of Operations

- Does the approach used in developing the ConOps reasonably ensure that the "system need" will be well understood and does not unduly constrain evaluation of alternative concepts?
- Are assumptions valid and reasonable?
- Does the evaluation criteria show a clear understanding of the system objectives?
- Does the proposed system fit within NARA's strategic goals?
- Are the benefits of the proposed system identified (such as improving existing processes or replacement of aging systems)?
- Does the ConOps present a feasible solution using established technology or has the proposed use of new technology been supported with appropriate justifications?
- Have alternative system concepts been expressed with a discussion of pros and cons?
- Does the recommended approach have a clear advantage over alternatives? (This should be supported with Cost-benefit analyses, Return-on-investment, or other comparable comparisons)
- Are logistics and operational support needs realistically addressed for the recommended approach?
- Does the recommended approach meet or exceed the evaluation criteria?
- If the system is to share data or integrate with the NARA architecture, does the Conceptual Data Model and Engineering Model illustrate those aspects?
- Has the Conceptual Data Model been developed in accordance with NARA standards?
- Has the 1st level Engineering Model been developed in accordance with NARA standards?
- Does the ConOps identify major technical risks associated with the recommended approach?
- Have the impacts to the organization and operations been addressed?
- Are technical cost estimates provided by functional area for evaluation purposes?
- Are cost estimates reasonably justified with market analysis, comparison of similar systems, or other acceptable basis?
- Do cost estimates fit the project allocation?

Presentation of Project Plan

Overview of project roles and expertise required to support the recommended approach
Work Breakdown Structure and Milestone schedule
Staffing and Training Plan
Total System Development Cost Estimate
Development Facilities Plan

Complete the following checklist for elements of the Project Plan

Schedules and Staffing checklist

- Has an appropriate level of expertise been defined to support the project?
- Has a schedule and WBS been created that corresponds to the Concept of Operations and the NARA Task Outline guidelines?
- Are activities and tasks defined according the assessment in the Planning Guidelines?
- Does the project staffing and training plan appear reasonable under current market conditions? (The plan should address risks with filling key positions and retention)
- Does training provide a clear benefit to the project or to the NARA organization?
- Can training be provided in a period that allows the project to meet deadlines?
- Do cost and resource estimates fit the project allocation?

Development Facilities Plan

- Has a development environment been defined that will support the project efforts?
- Has a work environment been defined that is maintainable for the life of the project?
- Does the development environment support the level of productivity needed? (Justification should be provided)

3.4 Concept Development Product Templates

<table>
<tr><th colspan="3">PROJECT PLAN TEMPLATE</th></tr>
<tr><th>Identification Number</th><th>Title</th><th>Purpose</th></tr>
<tr><td>NARA-IT-T-0001</td><td>Project Plan</td><td>The Project Plan describes how a specific system is to be developed including the specification of the systems engineering methodology and management lifecycle to be utilized and the integration of the technical management aspects with the planning, programming, and budgeting cycle and acquisition planning.</td></tr>
<tr><th colspan="3">Content Description</th></tr>
<tr><td colspan="3">The contents of a Project Plan should include the following sections and topics:</td></tr>
<tr><td colspan="3">1.0 Project Description.
This section will include the following topics:
Project Context.
This topic will set the context and scope of the project by referencing the governing version of the Product Plan and depicting the purpose of the project, a historical summary of how the project came into being, an overview of the current system that is to be updated or replaced, a graphic depicting the interfaces between projects/systems, and a description of the project goals. This topic could initially contain the project description summary included in the Product Plan, but may provide more detail over time. This discussion is not intended to replace the Concept of Operations Document, Requirements Document, or Statement-of-Work.
Project Plan Configuration Management.
This topic will describe the process for maintaining configuration control of the Project Plan and will include the naming convention and electronic file location of the most current plan.</td></tr>
<tr><td colspan="3">2.0 High-Level System Development Approach.
This topic will include a summary description of the system development lifecycle models and strategies selected for a given project and documented in the Product Plan. Specifically, this topic must include:

Development Lifecycle Model.
This topic will identify the steps to be followed in selecting the development lifecycle model, the selection, and the rationale for the selection. If the model deviates from the methodologies presented in the IT Solutions Guide, the rationale for the deviation must also be included.
Project Development Phases and Strategies.
Where the MED Model is used, the development strategies such as prototyping, piloting, and/or testbed evaluation must be selected and presented in a series of phases.</td></tr>
</table>

PROJECT PLAN TEMPLATE		
Identification Number	**Title**	**Purpose**
NARA-IT-T-0001	Project Plan	The Project Plan describes how a specific system is to be developed including the specification of the systems engineering methodology and management lifecycle to be utilized and the integration of the technical management aspects with the planning, programming, and budgeting cycle and acquisition planning.

Content Description

3.0 Detailed System Development Approach by Phase.

This topic will include the following:

Activity Identification.
This topic will identify the activities required for each system development phase.

Work Breakdown Structure.
This topic will describe the WBS in an activity hierarchy and/or infrastructure activity category to depict cost estimation and reporting.

Development Reviews/Audits Identification.
This topic will describe the required reviews/audits for each phase, tailoring considerations, and tailoring rationale.

Product/Deliverable Identification.
This topic will describe the required products/deliverables for each activity and phase, tailoring considerations, and tailoring rationale.

Acquisition Strategy Identification.
This topic will identify the steps to be followed in selecting an acquisition strategy, the selection, and the rationale for the selection. This is not meant to be a substitution for developing contract documents such as Statements-of-Work, Acquisition Plan, or Source Selection Plan.

4.0 NARA Infrastructure Requirements.

This section must include the following topics:

Risk Management.
This topic should include the list of technical and programmatic (cost and schedule) risks identified as part of the risk assessment activity. It should also contain risk assessment results and a risk mitigation approach for each of the identified risks.

System Engineering Diagramming Methodology.
This topic describes the plan to portray the system technical architecture using the NARA system engineering diagramming methodology.

Data Modeling.
This topic describes the plan to reconcile the Project/system data model with the NARA data model to facilitate data sharing, data distribution, and/or data migration.

Data Management.
This topic includes plans for data conversion, handling, and storage.

Configuration Management.
This topic describes the Project configuration management processes and procedures and where and when CM responsibilities transition between the contractor and NARA. It includes standards and procedures for configuration identification, configuration control, configuration status accounting, and configuration audits.

Quality Assurance.
This topic describes the process, procedures, and activities that will be executed to monitor, verify, and evaluate the Project's level of compliance with the contractual and technical requirements, and the technical standards and guidelines.

Security.
This topic will include a security plan describing activities and products for specifying system

PROJECT PLAN TEMPLATE		
Identification Number	**Title**	**Purpose**
NARA-IT-T-0001	Project Plan	The Project Plan describes how a specific system is to be developed including the specification of the systems engineering methodology and management lifecycle to be utilized and the integration of the technical management aspects with the planning, programming, and budgeting cycle and acquisition planning.
Content Description		

security including facility, hardware, software, and information security procedures, guidelines, controls, audits, and associated products.

Documentation Standards.

This topic will describe tailoring exceptions to the template standards provided in the IT Solutions Guide, rationale for template exceptions, and will reference waivers received as a result of concurrence with the Architecture Team.

Project Management and Controls.

This topic will include the agreement between the NARA Project Team and Senior Management regarding a periodic Project Management Review and Status reporting of the Project. Consideration should also be given to defining a process for elevating immediate or urgent Project issues to management. Metrics that are used to evaluate the system development progress should be identified and defined.

Test and Evaluation.

This topic area will include Test and Evaluation procedures and guidelines for acceptance of contractual products/deliverables.

Training.

This topic area will specify training requirements for the Project.

Project Library.

This topic will include the Project Team electronic directory structure for maintaining Project files including all versions of the Project Plan and the Project Deliverables. Consideration should be given to the possible need to promote electronic files for access by a wider audience, such as Senior Management and product evaluators and auditors, after approval. Document naming conventions should also be specified as well as directory access permissions.

5.0 Records Management.

This topic must document that the Project Team is aware of the requirement to integrate records management requirements into the system and to have the NARA Records Officer involved in the systems design process.

- Recordkeeping requirements associated with the system itself and the records it creates and/or maintains.
- Recordkeeping requirements associated with managing an IT project.

6.0 Software Development / Management Plan

This section must describe the software development approach to be used on the project and the methods for evaluating and controlling the software process. Items identified in the NARA Software Engineering Guidelines should be addressed. If the project is of sufficient complexity, this item may be expanded into a separate document.

7.0 Test Strategy

This topic should address the testing approach to be used during the project. Specifically, areas regarding the type of testing to be conducted (e.g., unit, component, system), how the testing will be evaluated for successful completion, and the expected level of effort to be applied to testing. The level of effort should be justified with the expected risk of system failure or defects.

8.0 Contractor Management.

This topic will include requirements for Contractor project management activities and products. If multiple contractors are part of the project, their individual requirements must be specified.

PROJECT PLAN TEMPLATE		
Identification Number	**Title**	**Purpose**
NARA-IT-T-0001	Project Plan	The Project Plan describes how a specific system is to be developed including the specification of the systems engineering methodology and management lifecycle to be utilized and the integration of the technical management aspects with the planning, programming, and budgeting cycle and acquisition planning.
Content Description		
9.0 Project Resources. This section will include the roles and responsibilities of the Project Team members. It will also include the NARA-organizational and contractor coordination interfaces.		
10.0 Project Schedule. This section will include the schedule of the Project to include the lifecycle model phases and activities and the major review milestones. The development schedule must be integrated with the planning, programming, and budgeting process and the acquisition schedule.		
11.0 Project Cost. This section will include the cost estimate of the Project by development Phase and by Fiscal Year. The cost will be categorized by hardware, software, personnel, and contractor costs.		

CONCEPT OF OPERATIONS TEMPLATE		
Identification Number	**Title**	**Purpose**
NARA-IT-T-0002	Concept of Operations	The Concept of Operations Document (ConOps) describes a proposed system in terms of the user needs it will fulfill, its relationship to existing procedures, and the ways it will be used. It is developed during the Concept Definition activity and is formally accepted at a Concept Review (CR).
Content Description		
The contents of the Concept of Operations should include the following topics:		
Introduction. The introduction sets the context of the document by identifying the Product Plan, Project Plan, and system to which it pertains, the background of the project, the scope of the development effort, an overview of the system, operational policies and constraints, document overview, relationship to other documents, and referenced documents.		
Approach. Include a description of the general approach or methodology being used for analysis and development of the operational concept. A brief explanation of any terminology or diagramming techniques used with the approach may be appropriate for assisting readers in understanding of the document. Also include the approach used for cost estimation, cost-benefit analysis, or any other supporting information as necessary.		
Assumptions. Identify any assumptions being used in preparing the system concept. These are particularly necessary if assumptions are being used for bounding the scope of the system, but the issues or problems involved with the boundaries are not yet well defined.		
Proposed System Description. This topic should describe the system mission or objectives, scope, high-level functional and data description, states or modes of operation, types of users of the system, and maintenance and support concepts. Subjects to address include the operational environment; interfaces to external systems or procedures, capabilities and functions of the system; performance characteristics such as speed, throughput, volume, and/or frequency; quality attributes such as reliability, maintainability, availability, flexibility, portability, usability, and efficiency; and provisions for security, privacy, and continuity of operations in emergencies. User descriptions should include organizational structures, training/skills, responsibilities, activities, and interaction with one another. Support concepts should include a description of support organizations, facilities and equipment, support software, repair/replacement criteria, maintenance level and cycles, and storage, distribution, and supply methods.		
Justification for and Nature of Changes. This topic should describe new or modified aspects of user needs, missions, objectives, environments, interfaces, personnel, or other factors that require a new or modified system. It should summarize deficiencies or limitations in the current system or situation that makes it unable to respond to these factors.		

CONCEPT OF OPERATIONS TEMPLATE		
Identification Number	**Title**	**Purpose**
NARA-IT-T-0002	Concept of Operations	The Concept of Operations Document (ConOps) describes a proposed system in terms of the user needs it will fulfill, its relationship to existing procedures, and the ways it will be used. It is developed during the Concept Definition activity and is formally accepted at a Concept Review (CR).

Content Description

Concept for a New or Modified System.

Operational Environment.

This topic should describe the new or modified system including as applicable, the operational environment and its characteristics, major system components and the interconnections among these components, and interfaces to external systems or procedures. The reach, range, and maneuverability (or interoperability) concepts should be illustrated using the System Engineering Level 1 Diagram.

System Engineering Level 1 Diagram (SED1).

The system engineering guidelines provide information for what should be included in the level 1 diagram.

Functional and Data Description.

The concept must also include a description of the functions and data to be used by the new or modified system. To help illustrate this, a Conceptual Data Model (CDM) should be included. A process, workflow, or data flow model may also be used to illustrate the user's point of view. These models can be used to describe operational scenarios that illustrate the interaction of the users with the system.

Conceptual Data Model (CDM).

See the NARA Data Modeling Guidelines for definition, specific requirements, and data modeling naming standards and conventions pertaining to the CDM.

Conceptual Process Model (CPM).

A process outline should be included in an easily understood format. This model is likely to be created in accordance with the project's development methodology.

Other System Requirements.

Additional descriptions may include performance characteristics, such as speed, throughput, volume, and frequency, quality attributes such as reliability, maintainability, availability, flexibility, portability, usability, and efficiency.

Provisions for security, privacy, and continuity of operations in emergencies should also be considered.

CONCEPT OF OPERATIONS TEMPLATE		
Identification Number	**Title**	**Purpose**
NARA-IT-T-0002	Concept of Operations	The Concept of Operations Document (ConOps) describes a proposed system in terms of the user needs it will fulfill, its relationship to existing procedures, and the ways it will be used. It is developed during the Concept Definition activity and is formally accepted at a Concept Review (CR).

Content Description

Summary of Impacts.
This topic area should discuss anticipated operational, organizational, and developmental impacts on the user, acquirer, developer, and support organizations.

Operational Impacts.
Specific operational impacts may include changes in procedures, interfaces with computer operating centers, use of new data sources, changes in quantity, type, and timing of data to be input to the system, or changes in data retention requirements.

Organizational Impacts.
Organizational impacts may include modification of, addition, or elimination of responsibilities; need for training or retraining; or changes in number, skill levels, positions, or location of personnel.

Developmental Impacts.
Developmental impacts may include meetings/discussions regarding the new system, development or modification of databases, training, parallel operation of the new and existing systems; impacts during testing of the new system; and other activities to aid or monitor development.

Analysis of Proposed System.

Advantages.
This topic should include a summary of advantages to be obtained from the new or modified system. These might include new capabilities, enhanced capabilities, and improved performance, as applicable, and their relationship to the deficiencies described under the *Justification for and Nature of Changes* topic area.

Disadvantages/Limitations.
A summary of disadvantages and limitations should also be described. These might include degraded or missing capabilities, degraded or less-than-desired performance, greater-than-desired use of computer hardware resources, undesirable operational impacts, conflicts with user assumptions, and other constraints.

Trade-Off Analysis Description.
Each alternative should be described, the trade-offs among them, and rationale for the decisions reached.

Cost Estimate.
Specify a breakdown of the cost by major system component or similar level. Include a basis of how the cost estimate was reached to show a reasonable justification.

4 Requirements Definition

4.1 Purpose

The Requirements Definition Activity focuses on developing detailed system specifications for the end product. As such, it begins to refine the Concept of Operations and define expected systems behavior in terms of performance, capacity, data inputs and outputs, processing, etc.

From a project management perspective, the Requirements Definition activity will provide the inputs necessary to begin a detailed risk assessment and continued systems development planning. It is during this activity that the project scope should be closely examined to determine if resource allocation and project expectations have been assessed accurately. The PM should use this information in discussions with Product Owners and the Guidance Team to revise project and product plans as needed to ensure success.

4.2 Requirements Definition Task Outline

The following timeline figure shows the typical sequence of Requirements Definition tasks and work product creation.

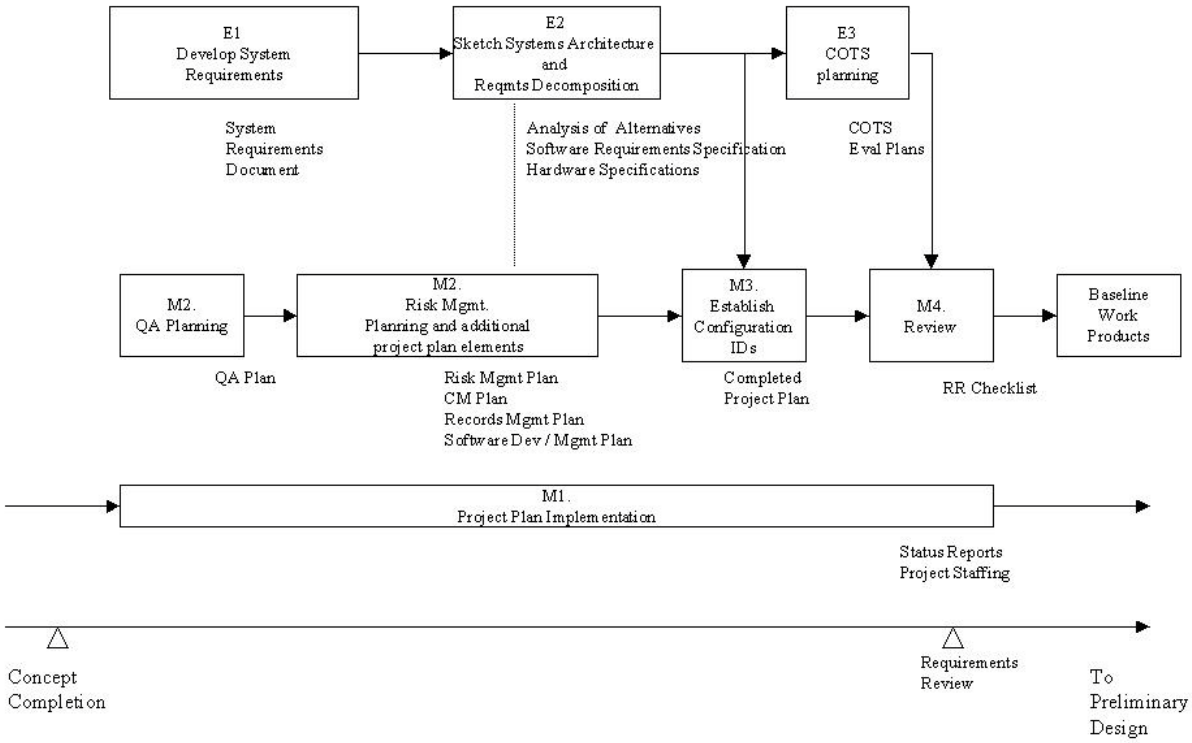

Figure 4-1 Requirements Definition Activity Timeline

Table 4-1 Requirements Definition Performance Requirements

Entry Criteria	**Requirements Definition Entry Criteria**
Products and activities to be completed before beginning a new software development activity.	Product Owner acceptance and baseline of ConOps Guidance Team approval of Project Plan elements and continuation to next activity Issues identified during the Concept Review have a documented resolution Detail schedule prepared for Requirements Definition activity
Work Products	**Requirements Definition Work Products**
Products and activities to be completed during the current software development activity.	Analysis of Alternatives - Recommended System Architecture Model - Component Trade-off analysis System Requirements Document - Functional, Data, and Records Management Requirements - System (External / Internal) Interface Requirements, - Security Functional and Assurance / Privacy Requirements - User and Operational Support Requirements - Software Requirements Specifications - Hardware Specifications - EA compliance requirements are identified and documented that include Business, oversight directives, mandates, etc. Outline of System Test Plan Fully Developed Project Plan - Quality Assurance (QA) Plan - Risk Management Plan including Security Risk Assessment - Configuration Management (CM) Plan - Software Development / Management Plan - Refinement of existing project plan elements
Exit Criteria	**Requirements Definition Exit Criteria**
What must be finished before an ongoing activity is designated complete	System Requirements Document(s) approved by Product Owner Refined Project Plan approved by Guidance Team Open issues from Requirements Review resolved and documented Detail schedule prepared for Preliminary Design activity Requirements and Project Plan documents are baselined

Engineering Tasks

E1. Develop System Requirements
- a. Refine the Concept of Operations to show systems behavior and timing (performance) needs.
- b. Perform Requirements (or Problem) Analysis in accordance with the project's development methodology. Techniques may include
  - i. Prototyping
  - ii. Use-case, activity, and class diagrams
  - iii. Functional or system flow diagramming
  - iv. N-square diagramming
  - v. etc.
- c. Create the system requirements specification
  - i. Functional / Performance / Reliability
  - ii. System interfaces (external, component, user)
  - iii. Logistical support
  - iv. Records and Data Management Requirements including data business rules and the Logical Data View and Data Models (LDV / LDM)
  - v. Environmental and other design constraints
- d. Identify any existing system and data migration requirements
- e. Develop outline of System Test Plan

E2. Outline systems architecture and derive system requirements to lower levels
- a. Allocate system functions to facilities, hardware, software, and other system components to establish Configuration Items
  - i. Prepare system models
  - ii. Develop justification for the proposed architecture
- b. Develop an Analysis of Alternatives for the proposed architectural components based on currently available technology
  - i. Perform reuse analysis
  - ii. Develop pros and cons of alternative approaches
  - iii. Establish Build vs. Buy criteria for the various components / configuration items
- c. Use the proposed System Architecture and Requirements specification to develop lower level hardware and software requirements
  - i. Develop software requirements in accordance with the methodology defined in the Software Development Plan
  - ii. Develop hardware, connectivity, and facility requirements in accordance with the Systems Engineering Guidelines

E3. Develop COTS Evaluation Plan(s) in accordance with NARA Standards
- a. Prepare both hardware and software evaluation criteria according to functional areas
- b. Identify candidate products

Management Tasks

M1. Implement Project planning for Requirements stage
- a. Monitor schedule
- b. Initiate project staffing and training
- c. Prepare development facilities
- d. Detail schedule for next activity

M2. Develop new project plan elements
- a. Prepare Quality Assurance plan
- b. Prepare Risk Management Plan
  - i. Conduct risk assessment
  - ii. Identify key requirements that have a strong influence on cost, schedule, functionality, or performance.
  - iii. Review significant risks to assess and negotiate project scope changes
  - iv. Document Risk factors, monitoring, and mitigation approach
- c. Prepare Configuration Management Plan
- d. Prepare Software Development / Management Plan
- e. Prepare Test Strategy
- f. Refine existing project plan elements based on new information

M3. Establish Release Identifications for defined Configuration Items
- a. Define item release numbers according to Configuration Management plan
- b. Prepare work areas, software sets, or other items as needed via the CM tools or according to CM plan

M4. Perform Requirements Review (RR)
- a. Prepare and Distribute RR materials (checklist, work products)
- b. Review materials prior to meeting
- c. Conduct RR meeting
- d. Resolve open issues
- e. Establish configuration baselines for work products

Roles	Responsibilities
Product Owner	• Provides clarification to development team during requirements development (E1 / E2) • Provides user and operator interface (training / support) requirements to development team(E1) • Assesses risk issues to determine project scope changes (M2) • Reviews and approves requirements specifications and systems architecture (M4) • Identifies records required by business need that the system should create and maintain
Guidance Team	• Assesses risk issues for project impact (M2) • Ensures adherence to NARA Architecture and Standards (M4) • Approves Project Plan revisions and start of next activity (M4)
Project Management	• Implements project plan (e.g., project staffing, monitoring schedule, documentation of meetings and agreements, etc.) (M1) • Prepares and/or incorporates new project plan elements (M2) • Performs Risk assessment and addresses significant issues with Product Owner and Guidance Team (M2) • Conducts Requirements Review (M4) • Completes RR checklist (M4)
Systems Engineering	• Prepares system interfaces, user profiles, security, facilities, and hardware requirements (E1) • Derives and allocates system requirements to functional components (presentation, processing, data management) (E1 / E2) • Defines Configuration Items (E2) • Ensures requirements traceability from system to lower level specifications (E2) • Prepares COTS Evaluation Plans (E3) • Ensures adherence to NARA system policies (E1 / E2 / E3) • Attends CM training (M1)
Data Management	• Prepares data business rules and Logical Data Model (E2) • Prepares data element mapping (interface requirements) between NARA and application data models (E2) • Prepares Data Access Lists (Security / User Profiles) (E2) • Provides input to COTS evaluation for data management and data access middleware components (E3) • Attends CM training (M1)

Roles	Responsibilities
Software Engineering	• Defines software development approach and environment e.g., object oriented vs. structured analysis, CASE tools, etc. for input to Software Development Plan and development facility needs (M1 / M2) • Prepares software requirements specification (E2) • Provides input to COTS evaluation for software application components (E3) • Attends CM training (M1)
CM	• Prepares CM plan (M2) • Implements CM library and associated tools (M3) • Provides training in use of CM tools (M3) • Prepares environment for development of work products in accordance with CM plan (M3) • Defines Release Identifications for current phase Configuration Items(M3) • Captures and baselines work products (M4)
QA	• Prepares QA plan (M1) • Assesses and approves adequacy of work products prior to Requirements Review (M4) • Ensures completion of RR checklist (M4)
Test and Evaluation	• Prepares project Test Strategy (M1) • Prepares outline of System Test Plan (E1) • Prepares Requirements Review evaluation checklist (M4) • Evaluates Requirements in accordance with Requirements Checklist (E1 / E2 / M4)
Operations and Maintenance	n/a

4.3 Requirements Review Checklist

Purpose and Objectives:

The Requirements Review is the formal review with Product Owners and the Guidance Team to review the set of detail system specifications prior to the design and development of a product. Specific objectives of the Requirements Review include

- Demonstrate that the product requirements are well understood and can be supported through defined system components
- Demonstrate that the project can be performed within resource allocation constraints
- Ensure that adequate detailed information exists to support initiation of further development or acquisition efforts.
- Inform the Product Owner and Guidance Team as to the risks associated with achieving successful completion of the project.

As the RR is a final formal review before execution of the Preliminary Design activity, it is expected that significant issues impacting project scope or resources have been identified and addressed during the Requirements Definition activity. *Use of the RR as a forum to address significant issues that could jeopardize the success of the project is not recommended.* The RR may, however, assess the outcome of known issues to make a decision on how to proceed.

Presenters

- Project Management
- Systems Engineering
- Data Management
- Software Engineering
- Test and Evaluation

Participants

- Product Owner(s)
- Guidance Team
- QA
- Operations and Maintenance

When:

The Requirements Review should be held after the Requirements Definition activity work products have been completed and QA has approved their readiness for review. Depending on the complexity of the project and deliverables, multiple reviews may be held so that an appropriate amount of time may be allocated to provide the review thoroughness necessary. One such break may be holding the technical requirements review and the project planning review on separate occasions. The Project Manager, Product Owner, and Guidance Team should make a determination on whether multiple reviews are warranted and the split that should occur.

Key Issues

- Is the project feasible, given the constraints on and assumptions about available resources?
- Have all critical requirements issues and technical risks been addressed?
- Is the defined system architecture an appropriate framework for satisfying the requirements?
- Is the foundation adequate to begin system design?
- Does the detail risk assessment create a need for changing the level of formality and control on the project (e.g., the risk assessment causes a project to go from medium risk to high risk)?

Entry Criteria

The following items should be done prior to conducting the review

Project Plan additions and modifications are distributed to participants 2 weeks prior to review Include:

- Proposed Schedule Revisions (should show baseline modifications)
- Proposed Resource Allocation changes

Requirements Documents are distributed to participants 2 weeks prior to review

COTS Evaluation Plans distributed 1 week prior to review

Exit Criteria

The review process is not considered closed until the following items are complete.

- All issues raised during the review have a documented resolution
- The Product Owner agrees and signs the Requirements Specifications
- The Guidance Team approves and signs the revised Project Plan (thereby agreeing to project continuation)
- QA confirms completion of the Review Checklist

Presentation of Requirements

General overview of approach to development of requirements and system architecture

Assumptions used in preparation of products

System Level Requirements

- Functional overview (including refined concept of operations) and requirements presentation
- Data and Records Management Requirements
- System and User Interface requirements
- Security and Privacy Requirements
- Environmental or other design constraints

Analysis of Alternatives / System Architecture

- Overview of system components / Systems Engineering Diagrams
- Relationship of architecture to system requirements
- Alternative approaches and reasons for rejection
- Technical considerations (limitations of existing technology, specific types of software or hardware required to support architecture, other assumptions that impact or limit design)

Software Specifications

- Software Architecture
- Software Component descriptions
- Data Business Rules, Logical Data Models and Data Views

Hardware Specifications

- Hardware, networking, and facility component descriptions
- Capacity and Performance specifications
- Connectivity specifications

COTS Evaluation Plans

- Product type descriptions / relation to requirements and architecture
- Candidate products overview
- Assessment plans

Complete the following Requirements Checklist

General

- Are assumptions used in preparation of requirements valid?
- Are all requirements testable and unambiguous?
- Are all requirements traceable from lower level to higher level (and ultimately the Statement of Need)?
- Have all critical requirements been specified i.e., is there a risk that system would not be designed adequately due to missing information?
- Are requirements classified as to their priority such as mandatory, "wish list", etc.?
- Are there any unnecessary requirements included in the specification i.e., could an unacceptable design be generated without the requirement?
- Is only desirable behavior specified?
- Have all requirements been resolved so that they do not conflict with each other?
- Has the source of derived requirements been adequately identified and documented?
- Are requirement statements written without design elements (e.g., statements are made like "data shall be available for transfer to external systems ... ", not "data shall be transferred with ftp over the Internet to external systems ...")

System Requirements

- Have all external data sources and systems been identified?
- Have user interface requirements been specified to include data needs and functional capabilities?
- Are records management requirements specified for system outputs?
- Have performance requirements been specified including system processing speed, system response time, system reliability and failure recovery time, and output data availability?
- Has security been addressed and appropriately balanced between access needs, cost, usability, and effort required? (System interfaces in particular should address this area.)
- Have any environmental constraints been factored into the requirements?
- Have logistical and operational support requirements been specified?

System Architecture

- Have all system level requirements been allocated to the system architecture?
- Does the system architecture support the requirements with a minimum level of complexity (i.e., are there any components that should be combined because their separation adds unnecessary complexity, should any components be split because their operations can be handled more efficiently through separation)? Evaluate factors such as portability, interoperability, and scalability.
- Do the components provide a reasonable separation such that the internal operations of one do not impact another (i.e., if the interface remains the same, does it matter what the internal operation is).
- Does the system architecture facilitate the identification of software and hardware Configuration Items?

Software

- Have requirements been defined in accordance with methodology in the Software Development Plan?
- Have data processing requirements been specified to show input, transforming processes, and output?
- Have behavioral requirements been specified for expected event responses, system states, etc.?
- Have common data entities and attributes from the NARA Data Model been identified?
- Are entities and attributes common with the NARA Data Model using the same definition?
- Does the Logical Data Model, Data Views, and Software Requirement Specifications address all applicable NARA business rules?

Hardware / Facilities

- Has any special-purpose hardware needs been identified e.g., for disabled access?
- Has data and processing capacity requirements been defined for each Hardware Configuration Item?
- Have presentation requirements been specified for hardware supporting user access (e.g., graphical capabilities)?
- Have interface requirements between system components been identified in terms of data transfer rates and other connectivity requirements?
- Have fault tolerances been specified?

COTS Evaluation Plans

- Does the plan correlate to one or more functional components of the System Architecture?
- Can COTS items be controlled as a Configuration Item?
- Have candidate products been identified that meet the high level product criteria outlined in the Evaluation Plan?
- Is selection criteria written in accordance with the Agency guidelines?

Project Plan Presentation

Assumptions used in construction of new project plan elements

Current project status against schedule and budget

Impact of new elements
- Revisions to existing project plan elements
- Project scope or resource changes
- Project risk level

New plan elements
- Risk Management Plan
- QA Plan
- CM Plan
- Software Development / Management Plans
- Test Strategy

Facility and Staffing status

Schedule of next activity

Complete the following Project Plan Checklist

General Project Plan elements

- ❑ Are assumptions used for new project plan elements valid?
- ❑ Is the project progressing per the schedule and within budget? If not, have proposed changes been submitted for review and approval?
- ❑ Have existing plan elements (staffing, environment) been implemented?
- ❑ Has the project plan and schedule been sufficiently detailed to allow continuation to the next activity?

Risk Management Plan

- ❑ Does the risk plan accurately portray potential risks on the project?
- ❑ Does each risk item have a mitigation plan that is practicable?
- ❑ Have adequate controls been implemented for monitoring and managing risks?

QA Plan

- ❑ Does the QA plan establish practices relative to the project risk level, development phase, and application type in accordance with the SDLC Planning Guidelines?

CM Plan

- Have all elements of the NARA Configuration Management guidelines been addressed?
- Are the methods for controlling changes to system products appropriate for the level of risk, development phase, and application type in accordance with the SDLC Planning Guidelines?
- (If necessary) Does the CM plan cover concurrent development paths for operational environments?

Software Development / Management Plan

- Does the plan cover all aspects identified in the NARA Software Engineering Guidelines?
- Is the development plan appropriate to the level of risk, development phase, and application type in accordance with the SDLC Planning Guidelines?

Test Strategy

- Does the strategy outline an approach for developer, system, and acceptance test levels?
- Is the testing to be performed on the project appropriate to the risk level?
- Is a method for assessing the testing effectiveness identified?
- Is an approach for tracking, managing, and reporting test incidents specified appropriate to the project risk level?

Proposed Schedule / Resource Revisions

- Have reasons for the schedule or resource changes been adequately tracked to the cause and documented?
- Do project revisions increase the risk category assigned to the project? If so, has the project taken appropriate steps to control the activities and implement the rigor necessary?
- Can the product mission be achieved with the proposed changes?

4.4 Requirements Definition Product Templates

REQUIREMENTS SPECIFICATION TEMPLATE		
Identification Number	**Title**	**Purpose**
NARA-IT-T-0004	Requirements Document	The purpose of the Requirements Document is to specify functional and data, external and internal interface requirements, security and privacy requirements, technical architecture requirements, training and personnel requirements, and quality factors for a specific system.
Content Description		
The contents of a Requirements Document should include the following topics:		
Introduction. The introduction sets the context of the document by identifying the Product Plan, Project Plan, and system to which it pertains including the identification number(s), title(s), abbreviation(s), version number(s), and release number(s), an overview of the system, and history of the system development, operation, and maintenance. It should also include the identification of the Project Team members including the Product Owner, acquirer, users, developer, and support organizations and identify current and planned operating sites. References to documents used in the generation of the document should also be specified as well as those documents with which it relates.		
Approach. Describe the approach or approaches used to analyze and development requirements. If a particular technique or diagramming method is used, briefly define terminology and notations.		
Assumptions. Indicate any current assumptions used in specification of requirements. This topic should be reviewed and revised during any change to requirements to assess validity.		

REQUIREMENTS SPECIFICATION TEMPLATE

Identification Number	Title	Purpose
NARA-IT-T-0004	Requirements Document	The purpose of the Requirements Document is to specify functional and data, external and internal interface requirements, security and privacy requirements, technical architecture requirements, training and personnel requirements, and quality factors for a specific system.

Content Description

Functional, Data, and Records Management Requirements.

The requirements must present a description of the known functions and data to be used by and allocated to the new or modified system. These three areas are closely related and generally occur in tandem.

To help illustrate system requirements, the data business rules, and a Logical Data View (LDV) and a Logical Data Model (LDM) should be included. Other materials that help illustrate the problem analysis or functional concepts of the system may be included here in accordance with the project's development methodology. Response times and other performance requirements associated with functional processing needs should be established during the development of this section.

Records Management requirements are to be defined for system record types and lifecycles. These are required to derive system storage capacity needs, backup requirements, etc. Record types should be defined for the transactions to be captured and maintained. In many cases the transactions will relate directly to the major processing functions defined for the system. Records management requirements are likely to provide input to the logical data views and models described below.

Logical Data View (LDV)

- A data dictionary of the data entities and relationships to be created, read, updated, or deleted by the application, as seen from a particular user or transaction perspective.
- Logical entity-attribute-relationship diagram(s) in First Normal Form.

Logical Data Model (LDM).

- A data dictionary of the data entities, attributes, and relationships to be created, read, updated, or deleted by the application.
- A Third Normal Form (3NF) logical entity-attribute-relationship diagram.

System External Interface Requirements.
This topic should describe the system interface requirements, if any, with inter- or intra- agency systems.

System Internal Interface Requirements.
This topic should describe the interface requirements between internal system components.

Security and Privacy Requirements.
This topic area should describe security requirements pertaining to physical and automated access to the system and system data.

Other Requirements.
This topic area should describe other requirements not covered in previous topics including user and operational support requirements; also specific system quality factors such as reliability, availability, maintainability, flexibility, portability, reusability, testability, usability, and other attributes.

REQUIREMENTS SPECIFICATION TEMPLATE		
Identification Number	**Title**	**Purpose**
NARA-IT-T-0004	Requirements Document	The purpose of the Requirements Document is to specify functional and data, external and internal interface requirements, security and privacy requirements, technical architecture requirements, training and personnel requirements, and quality factors for a specific system.
Content Description		
Technical Architecture Requirements. This topic area should describe the target architecture requirements in accordance with interoperability standards and requirements. *This is not the architecture itself.* Architecture topics include System Environment, Hardware, Computer Resource Utilization, and Communications.		
Software Requirements. This topic area should describe the requirements for each defined software application or component. It should be written in accordance with the methodology being used for the project as defined in the Software Development Plan. If the project is of sufficient complexity, this topic may be created as a separate document.		

ANALYSIS OF ALTERNATIVES TEMPLATE		
Identification Number	**Title**	**Purpose**
NARA-IT-T-0005	Analysis of Alternatives Document	An Analysis of Alternatives Document describes potential automated solutions, compares them against the business and technical problem statement or need, and provides a recommended solution or set of solutions with supporting rationale.

Content Description

The contents of an Analysis of Alternatives Document should include the following topics:

Introduction.
The introduction sets the context of the document by identifying the Product Plan, Project Plan, and system to which it pertains, the background of the project, the scope of the development effort, an overview of the system, operational policies and constraints, document overview, relationship to other documents, and referenced documents.

Problem Description.
This topic should describe new or modified aspects of user needs, missions, objectives, environments, interfaces, personnel, or other factors that require a new or modified system. It should summarize deficiencies or limitations in the current system or situation that make it unable to respond to these factors.

Alternative Solution Description.
This topic should summarize the capabilities of each potential solution. The summaries should consider the user-perspective as well as the technical architecture perspective.

Comparison Analysis.

Advantages.
This topic should include a summary of advantages to be obtained from each alternative system. These might include new capabilities, enhanced capabilities, and improved performance, as applicable, and their relationship to the deficiencies described in the *Problem Description*.

Disadvantages/Limitations.
A summary of disadvantages and limitations should also be described for each alternative. These might include degraded or missing capabilities, degraded or less-than-desired performance, greater-than-desired use of computer hardware resources, undesirable operational impacts, conflicts with user assumptions, and other constraints.

Recommendation.
This should provide a recommended choice or group of choices based upon the comparison analysis. Rationale for the selection should be documented. A basic systems architecture diagram (e.g., a block diagram or an outline for level 2 engineering diagrams) should be provided for use in developing COTS plans and lower level requirements.

5 Preliminary Design

5.1 Purpose

The Preliminary Design tasks begin to address how the Requirements will be fulfilled e.g., what COTS packages will be used and what requirements will they meet, what hardware and operating systems will support the system architecture, what will the user interfaces look like and how will they function, etc. Moreover, and of critical importance, is the definition of how these various items will work together.

The management aspects of this activity must be concerned with identifying and negotiating changes to requirements. As the design is developed, the requirements may need clarification or changes to support an efficient design. The impact of changes need to be assessed for whether the system will continue to support its mission, effect on resources, and any permutations in related components.

The following timeline figure shows the typical sequence of Requirements Definition tasks and work product creation.

5.2 Preliminary Design Task Outline

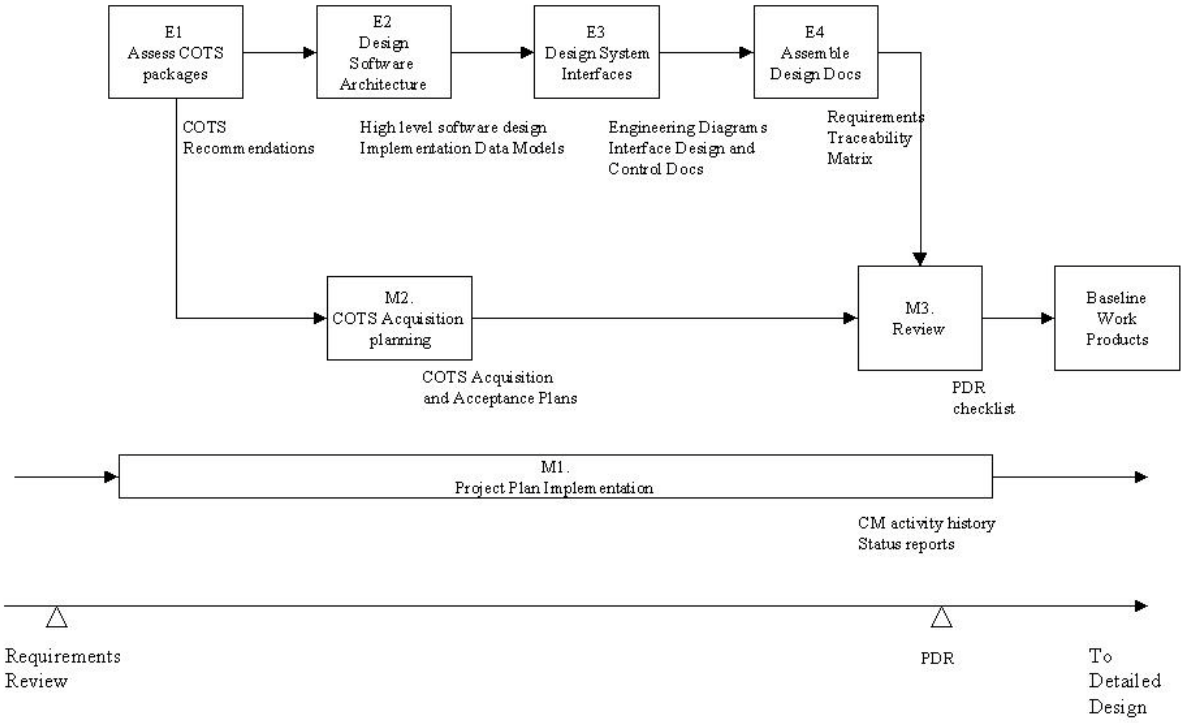

Figure 5-1 Preliminary Design Activity Timeline

Table 5-1 Preliminary Design Performance Requirements

Entry Criteria	**Preliminary Design Entry Criteria**
Products and activities to be completed before beginning a new software development activity.	System Requirements Document(s) approved by Product Owner Project Plan approved by Guidance Team Open issues from Requirements Review resolved and documented Detail schedule prepared for Preliminary Design activity
Work Products	**Preliminary Design Work Products**
Products and activities to be completed during the current software development activity.	Preliminary Design Document(s) including - COTS recommendations - Data Model Implementation Strategy and Data Implementation Strategy - High-level software design - 2nd level Systems Engineering Diagrams - Interface Designs and Control Documents - Records Management Requirements - Draft records schedule for the system initiated - Requirements Traceability Matrix - Compliance with NARA standards (See NARA EA, Appendix B – Technology Standards Profile) - Architectural design conforms to EA principles, guidelines, standards, and specifications - Exceptions to EA specification are documented as waiver requests and approved. COTS Acquisition and Acceptance Test Plans
Exit Criteria	**Preliminary Design Exit Criteria**
What must be finished before an ongoing activity is designated complete.	High Level Design approved by Product Owner Requirements and resource changes approved by Guidance Team Issues from Preliminary Design Review resolved and documented Detail schedule for Detailed Design activities (including Records Schedule) Preliminary Design documents are baselined

Engineering Tasks

E1. Perform COTS Analysis
   a. Assess impact to the draft Systems Architecture based on availability of COTS software / hardware
   b. Refine the draft Systems Architecture based on recommended COTS items
   c. Prepare COTS recommendations per guidelines

E2. Design the Software Architecture in accordance with NARA Standards
   a. Develop Functional (e.g., structure charts, flow charts), Component and Deployment diagrams, or other Diagrams per the project's methodology
   b. Define COTS implementation
      i. Customization needs
      ii. System configurations
   c. Design User Interfaces and Help systems

E3. Design system infrastructure
   a. Develop 2nd level engineering models
   b. Prepare Data Model Implementation Strategy and Data Implementation Strategy
   c. Model expected network and hardware configurations
   d. Prepare Interface agreements (MOUs, Interface Control Documents, etc.)
   e. Design testing environments

E4. Prepare Preliminary Design Documentation
   a. Collate and organize designs
   b. Construct a Requirements Traceability Matrix

Management Tasks

M1. Implement Project Plan elements specified for the Design Phase
   a. Implement Configuration Management procedures (e.g., Configuration Control Board)
   b. Implement QA and Risk management practices
   c. Monitor schedule / develop detail for next activity
   d. Staffing and training
   e. Generate status reports

M2. Initiate COTS acquisition
   a. Develop COTS acceptance test plans

M3. Perform Preliminary Design Review (PDR)
   a. Prepare and Distribute PDR materials (checklist, work products)
   b. Review materials prior to meeting
   c. Conduct PDR meeting
   d. Resolve open issues
   e. Establish configuration baselines for work products

Roles	Responsibilities
Product Owner	• Approves requirement changes through CCB (M1) • Approves preliminary design (M3)
Guidance Team	• Approves schedule and resource allocation changes (through CCB), COTS acquisition, and continuation to next activity (M1 / M3) • Ensures adherence to NARA Architecture and Standards (M3)
Project Management	• Implements project plan elements (CM, QA, Risk) (M1) • Facilitates CCB (M1) • Assesses impact of proposed requirements changes on schedule / budget (M1) • Prepares status reports (M1) • Develops COTS acceptance plan (M2) • Conducts PDR (M3) • Completes review checklists (M3)
Systems Engineering	• Evaluates COTS items to ensure compatibility between system components (E1) • Prepares level 2 SE diagrams (E3) • Prepares interface agreements for external systems (E3) • Reviews adherence of designs to NARA architecture standards or assesses impact of changes (E1 / E2 / E3) • Assembles Design Documents / Creates Requirements Traceability Matrix (E4) • Prepares technical impact assessment of requirements changes (M1) • Provides inputs to procurement requests (M2)
Data Management	• Evaluates COTS items for data management applications and access middleware (E1) • Prepares Data Model Implementation Strategy and Data Implementation Strategy (E2) • Designs data sharing procedures and functions between the project's application(s), external systems, and the NARA data repository (E2 / E3)
Software Engineering	• Evaluates COTS software components for presentation and processing layers (E1) • Designs User Interfaces, help systems, and data processing components (E2) • Assists in design of data sharing procedures and other system interface components (E2 / E3)
CM	• Implements the Change Request and Problem Reporting system in accordance with the CM plan (M1) • Provides training in use of CR / PR system (M1)

Roles	Responsibilities
	• Controls revisions to baseline documents i.e., manages the CM library (M1) • Documents change history (M1) • Captures and baselines work products (M3)
QA	• Validates work products in accordance with QA plan (M1) • Assesses and approves adequacy of work products prior to Preliminary Design Review (M3) • Ensures completion of PDR checklist (M3)
Test and Evaluation	• Evaluates COTS testing tools (E1) • Designs the system testing environment(s) (E3) • Provides input on COTS acceptance test plans (M2) • Identifies applicable elements of PDR checklist (M3)
Operations and Maintenance	• Identifies the operational procedures necessary to support the design (E2 / E3)

5.3 Preliminary Design Review Checklist

Purpose and Objectives:

The PDR is conducted to evaluate the feasibility of a proposed design to meet the product's requirements. It also provides an early assessment of whether the project is adequately progressing to provide an end product.

Objectives for the PDR include:

- Gain concurrence among development team, Product Owner, and Guidance Team that a system design has been defined that will adequately address all requirements.
- Demonstrate that the design avoids undue complexity.
- Ensure that progress is being made within the resources allocated to the project and that risks are being managed effectively.

The PDR is the first milestone where changes to baseline requirements for the current release may have occurred. This review may assess whether changes have impacted the scope or risks associated with the project. While significant changes may have been discussed previously, there may be a cumulative effect of smaller issues that is not evaluated until the PDR is conducted.

Presenters

- Project Management
- Systems Engineering
- Software Engineering
- Data Management
- Test and Evaluation

Participants

- Product Owner(s)
- Guidance Team
- Developers
- QA
- Operations and Maintenance

When:

After tasks for the Preliminary Design activity are complete up to the review. Multiple reviews may be held if appropriate, but should be split so that complete Configuration Items may be evaluated in their entirety.

Key Issues

- Is the project progressing in accordance with the schedule? If not, have appropriate actions been taken to realign resources and expectations?
- Is the schedule for the next activity reasonable in light of the preliminary design?
- Does the design support the system requirements without adding unnecessary risk?
- Have alternative designs been sufficiently evaluated?
- Has design been sufficiently developed to allow progress to Detailed Design activity?
- Has the NARA Records Officer been involved in the records scheduling process?

Entry Criteria

The following items should be distributed 2 weeks prior to conducting the review

Design documents
Summary report of Requirements changes
Procurement Requests
Schedules for future activity

Exit Criteria

The review process is not considered closed until the following items are complete.

- All issues raised during the review have a documented resolution
- The Product Owner agrees to the Preliminary Design
- The Guidance Team approves resource and schedule changes (thereby agreeing to project continuation)
- QA confirms completion of the Review Checklist

Presentation of Preliminary Design

Summary of changes to requirements caused by design

Overview of design approaches

Assumptions used in preliminary design

Hardware / Network / Facilities Design (Systems Architecture Design)

- System Components (2nd level systems engineering diagram)
  - Servers / Workstations / Operating Systems
  - Storage systems
  - Network topology / layout
  - Facility components (Space design, Power systems and backups, etc)
- COTS selection / configuration / customization
- Discussion of Alternative approaches
- Requirements Traceability Matrix

External Interface Design

- Descriptions of interfaces
- Means of data exchange between systems
- MOUs, Data Sharing Agreements, and Interface Control Documents

Software Design

- Software Components
  - User Interface Components
  - External System Interface Components
  - Data Processing Components
  - Application Programming Interfaces (API)
  - Data Model Implementation Strategy
- COTS selection / configuration / customization
- Discussion of Alternative approaches
- Requirements Traceability Matrix

Software Design Checklist

General Design

- Are all assumptions valid?
- Have software components been defined that support all software requirements?
- Have components been designed in accordance with the approach defined in the Software Development Plan?
- Are software components loosely coupled i.e., does the design support the easy replacement of components as long as the interface remains the same?
- Have software units been decomposed such that they tend to be of maintainable size and complexity e.g., do they avoid being overly large or too small to be useful? (Size and complexity may be measured in a number of ways such as estimated lines of code, function points, fan-out, etc.)
- Is error handling consistent throughout the design?
- Does the design support the testing approach defined in the Test Strategy?

User Interfaces

- Are user interface designs intuitive i.e., can processes be initiated with a minimum number of actions required by the user?
- Does the interface make it easier for the user to perform their job?
- Have processes been designed to be independent of user actions? (Avoid designs that lock records or processes while waiting for an external event unless there is a particular reason to do so.)
- Have help functions been defined?

System Interfaces

- Have data exchange processes been defined?
- Have processes been designed to be independent of external actions? (Avoid designs that lock records or processes while waiting for an external event unless there is a particular reason to do so.)
- Have data transformation procedures been defined for those that have to be manipulated between systems?

Data Processing Components

- Are data processing sequences identified that must be monitored and managed to ensure data integrity (i.e., "logical transactions" or "units of work")?
- Have the mechanisms been defined for monitoring each transaction?
- Are recovery checkpoints or rollbacks defined for failed transactions? (The design must provide for maintaining data integrity)
- Are functions self-contained so that they do not depend on external events for completion?
- Have other data integrity issues been addressed such as domain edits or duplicate transactions? (This may be partially addressed in the data models)

Component Interfaces / APIs

- Have custom interfaces been designed for minimal changes? (The interface should be implementation independent. Also use of parameters or polymorphism is preferred over the need to create a new interface.)
- When using COTS or 3rd party APIs, are they based on an open architecture (interoperable, scalable, and portable)?
- Is the use of a proprietary API justified if an open standard is available? (Example: ODBC or JDBC vs. Oracle Call Interface)

Data Modeling

- Does the Data Model Implementation Strategy and Data Implementation Strategy address NARA Data Deliverable guidelines?
- Are entities and attributes common with the NARA Data Model using the same definition?
- Does the system architecture support the Implementation Model and Strategy?
- Does the Data Model Implementation Strategy avoid any overly complex distribution of data? (Distributed databases should have clear benefits when it may be difficult to synchronize or replicate data.)

COTS

- Will the recommended products support and adapt to evolving business processes?
- Have COTS customization needs been identified and addressed in the design?
- Does each recommended product have a justification?
- Does the Systems Architecture support integration of the COTS item?
- Do selected products use a common interface, or open architecture, that allow them to be easily replaced?

Hardware / Network / Facility Design Checklist

General

- Does the system design fit within the NARA target architecture?

Processing platforms

- Do the processing platforms support the expected capacity for the life of the system or provide an easily identifiable upgrade path (i.e., is it scalable)?
- Will the processing platforms support the software and devices that are expected to run on them (e.g., will the operating system support the software and network protocols)?

Workstations / Interface Devices

- Will the workstations or other devices support all identified user presentation requirements?

Storage Systems

- Do the storage systems have a defined means for backup and recovery of data?
- Is the capacity and performance appropriate to the life of the system or provide an upgrade path?
- Do the storage systems provide an easy path for migration or replacement of media?

Network

- Will the network design support all data transfer needs?
- Does the network support the necessary level of access security?

Facility

- Will the facility design support all operational and logistical requirements such as work space, security, power outages, etc.?

Project Management Presentation

Summary report of requirements changes
Impacts to Schedules / Resources
Current project status against schedule and budget
Risk Mitigation efforts
Schedule for future activities

Project Management Checklist

- Are actual vs. estimated items on target? If not, have actions to correct schedule slippage or cost overruns been implemented?
- Is the schedule for future work realistic?
- Have project risk factors been actively monitored?
- Have risk mitigation efforts been implemented where required?

Proposed Schedule / Resource Revisions

- Have reasons for the schedule or resource changes been adequately tracked to the cause and documented?
- Do project revisions increase the risk category assigned to the project? If so, has the project taken appropriate steps to control the activities and implement the rigor necessary?
- Can the product mission be achieved with the proposed changes?

5.4 Preliminary Design Product Templates

PRELIMINARY SYSTEM DESIGN TEMPLATE		
Identification Number	**Title**	**Purpose**
NARA-IT-T-0006	Preliminary System Design Document	The purpose of a Preliminary System Design Document is to specify the system architecture design (hardware configuration), external interface design, software design, and database design of a specific system.
Content Description		
The contents of a Preliminary System Design Document should include the following topics:		
Introduction. The introduction sets the context of the document by identifying the Product Plan, Project Plan, and system to which it pertains, the background of the project, the scope of the development effort, an overview of the system, operational policies and constraints, document overview, relationship to other documents, and referenced documents.		
Approach. Describe the approach or approaches used in the design. If a particular technique or diagramming method is used, briefly define terminology and notations.		
Assumptions. Identify any assumptions being used in creating the preliminary design. Risks associated with design assumptions should be highlighted as to their potential impact and mitigation.		
System Architecture Design. This topic should describe the system components, configuration, and internal interfaces. It should use a Systems Engineering Level 2 Diagram (SED2) to depict the architecture design specifications. See the Systems Engineering Modeling Guidelines for a definition and specific requirements pertaining to the SED2. Of particular importance, this section should describe how any COTS items will interact with each other or integrate with custom components.		

PRELIMINARY SYSTEM DESIGN TEMPLATE		
Identification Number	**Title**	**Purpose**
NARA-IT-T-0006	Preliminary System Design Document	The purpose of a Preliminary System Design Document is to specify the system architecture design (hardware configuration), external interface design, software design, and database design of a specific system.

Content Description

External Interface Design.
This topic should describe the design for the interfaces with external systems. A Systems Engineering Level 2 Diagram should be used to depict the architecture interface specifications. Interface descriptions should include the standard data element name, technical name, abbreviation or synonymous names, data types, size and format, units of measurement, range or enumeration of possible values, accuracy and precision, priority, timing, frequency, volume, sequencing, and other constraints, security and privacy constraints, and sources and recipients.

In addition, interface descriptions should include the characteristics of data element assemblies (records, messages, files, arrays, displays, reports, etc.) that the interfacing entities will provide, store, send, access, receive, etc such as the identifier, data elements in the assembly and their structure, medium, visual and auditory characteristics of displays and other outputs, relationships among assemblies such as sorting/access characteristics, priority, timing, frequency, volume, sequencing, and other constraints, such as whether the assembly may be updated and whether business rules apply, security and privacy considerations, and sources and recipients.

Descriptions of communications methods should include communication links/bands/frequencies/media and their characteristics, message formatting, flow control, data transfer rate, routing, addressing, and naming conventions, transmission services, and security and privacy such as encryption, user authentication, compartmentalization, and auditing.

Protocol descriptions should include priority/layer of protocol; packeting, including fragmentation and reassembly, routing, and addressing; legality checks, error control, and recovery procedures; synchronization, including connection establishment, maintenance, termination; status, identification, and any other reporting features.

Descriptions should include other characteristics such as physical compatibility of the interfacing entities (dimensions, tolerances, loads, voltages, plug compatibility, etc.)

Software Design.
This topic should describe the specific software application. It should include design decisions regarding inputs the application will accept and outputs it will produce, including interfaces with other systems. It should include design decisions on application behavior in response to each input or condition, including the actions the application will perform, response times and other performance characteristics, description of the physical systems modeled, selected equations/algorithms/rules, and handling of unallowed inputs or conditions.

User Interfaces and associated descriptions (including data requirements, event processing, etc.) should be included under this topic.

All Software Design should be documented according to the methodology described in the project's Software Development Plan.

PRELIMINARY SYSTEM DESIGN TEMPLATE		
Identification Number	**Title**	**Purpose**
NARA-IT-T-0006	Preliminary System Design Document	The purpose of a Preliminary System Design Document is to specify the system architecture design (hardware configuration), external interface design, software design, and database design of a specific system.

Content Description

Database Design.

Preliminary Design should include creation of the Data Implementation Strategy (DIS) and the Data Model Implementation Strategy. These are created according to the NARA Data Deliverables guidelines.

Data Implementation Strategy includes A report outlining the following strategies:

- Data Sharing - Identification of data common to other applications, and approach for ensuring that changes made to any common field in any database are properly replicated in corresponding fields in other databases. If some of the data is common but incompatible due to format or domain reasons, describe how the data will be transformed (for example, by middleware).
- Data Distribution - If there are performance / cost reasons to distribute the data, this describes the strategy for partitioning and maintaining the integrity of the replicated data.
- Data Migration - If there will be legacy data or data from external sources used to initially populate the database, this describes what approach will be taken to bring that data in and especially how quality assurance of the data (cleansing, etc.) is done prior to the migration.
- Data Security/Sensitivity/Access - If any of the data is classified or subject to privacy laws, deed restrictions, etc., this describes the approach being taken to ensure these requirements are physically met.

Data Model Implementation Strategy includes

- Textual documentation describing reasons for each denormalization (e.g., logical transaction data access path mappings, volumetrics, etc.) or change from the Logical Data Model to the Physical Data Model.

6 Detailed Design

6.1 Purpose

The Detailed Design tasks finalize all the logical details of the system before full-scale construction of the system begins. In conjunction with design of logical units, developer test plans for each process are defined. Unit and component test planning at this stage will provide a checks and balances mechanism to ensure individual logical units and components are not overly complex.

Management during the detailed design activity will focus on controlling changes to baseline products and resolving conflicts among the various system interfaces. COTS items are acquired during this activity and undergo acceptance testing upon delivery. It is during this activity that a fairly deep understanding of the system to be constructed will be formed and this should lead to the integration and system test plans being developed in final draft form so that they may be reviewed during the CDR.

6.2 Detailed Design Task Outline

The following timeline figure shows the typical sequence of Detailed Design tasks and work product creation.

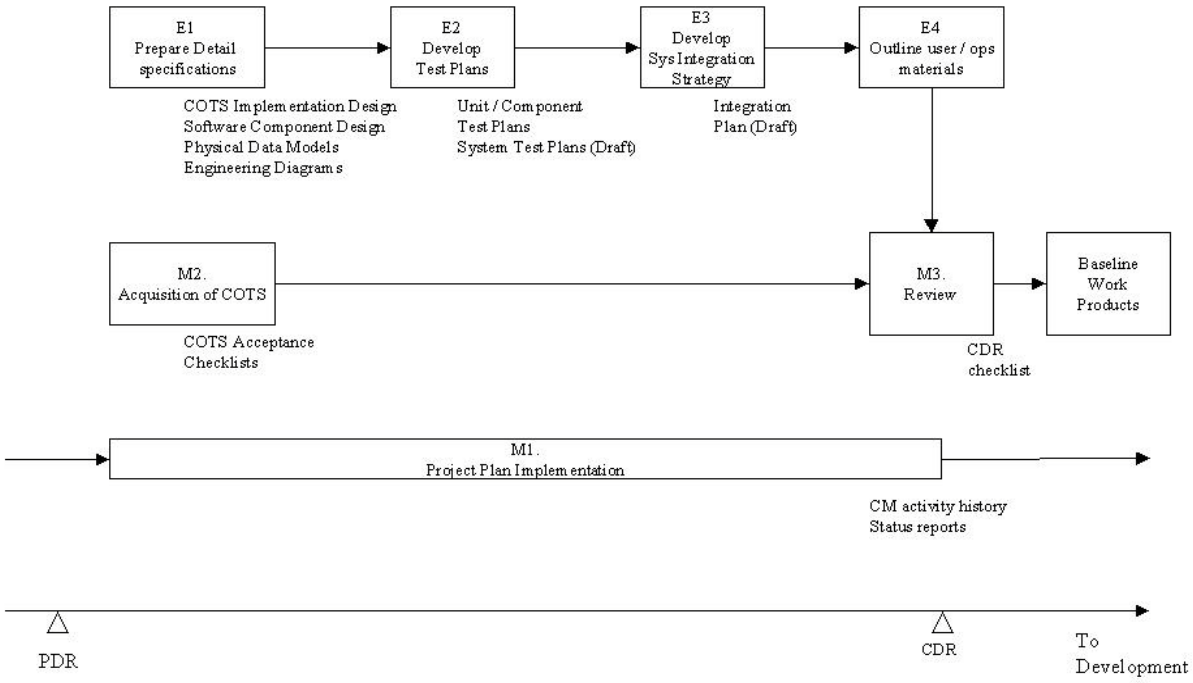

Figure 6-1 Detailed Design Activity Timeline

Table 6-1 Detail Design Performance Requirements

Entry Criteria	**Detail Design Entry Criteria**
Products and activities to be completed before beginning a new software development activity.	High Level Design approved by Product Owner Requirements and resource changes approved by Guidance Team Issues from Preliminary Design Review resolved and documented Detail schedule for Detailed Design activities
Work Products	**Detail Design Work Products**
Products and activities to be completed during the current software development activity.	Detailed Design Specifications - COTS Implementation Details - Software Component Design - Physical Data Models - 3rd Level System Engineering Models - Conform to the principles, guidelines, standards and specifications provided by the Data Architecture, Application Architecture, Systems Architecture, Security Architecture, and Operations Architecture from the EA Unit / Component Test Plans System Test Plan (Draft) System Integration Plan (Draft) User Guide and Operations Manual (Outlines) Security Plan and Security Control Development Developmental Security Test & Evaluations
Exit Criteria	**Detail Exit Criteria**
What must be finished before an ongoing activity is designated complete.	Detailed design approved by Product Owner and Guidance Team Resource allocation changes approved by Guidance Team Detail schedule prepared for Development and Integration activity

Engineering Tasks

E1. Develop Detailed Design Specifications
   a. Define COTS automation procedures or other implementation needs
   b. Create algorithm flowcharts, PDL, object specifications, or other detail logical models per the project's software development methodology
   c. Create software component API specifications / message exchanges
   d. Review software components and ensure APIs match between specification and usage
   e. Define Physical Data Model in accordance with NARA Data Deliverable guidelines
      i. Define triggers / stored procedures necessary to maintain data integrity and business rules
      ii. Define user groups and data access mechanisms
      iii. Define any other configuration elements for ensuring performance and data administration requirements are met
   f. Create 3rd level engineering models
   g. Prepare facility and hardware configuration models

E2. Develop Test Plans
   a. Prepare unit and component test plans in accordance with project Test Strategy
   b. Update the System Test Plan outline to create a draft version for review

E3. Develop System Integration strategy
   a. Define configuration item delivery procedures
   b. Define software build procedures
   c. Define system deliverables baseline strategy
   d. Define system initialization procedures

E4. Prepare user and operator materials
   a. Create user guide and operations manual outlines
   b. Identify training approach and likely material requirements

Management Tasks

M1. Design phase project plan elements
   a. Control changes to requirements / design according to CM plan
   b. Perform risk monitoring / mitigation
   c. Perform QA tasks
   d. Monitor schedule / Prepare detail for next activity
   e. Prepare status reports

M2. COTS acquisition

M3. Perform Critical Design Review (CDR)
   a. Prepare CDR materials (checklist, work products)
   b. Review CDR materials prior to review
   c. Conduct CDR
   d. Resolve open issues
   e. Establish configuration baselines for work products

Roles	Responsibilities
Product Owner	• Approves changes to requirements and high-level design through CCB (M1) • Approves detailed design (M3)
Guidance Team	• Ensures adherence to NARA standards • Approves schedule / budget changes (possibly through CCB) and continuation to next activity
Project Management	• Facilitates CCB (M1) • Assesses impact of proposed changes on schedule / budget (M1) • Implements project plan design phase elements (M1) • Initiates COTS acquisition (M2) • Conducts CDR (M3) • Completes CDR checklist (M3)
Systems Engineering	• Prepares 3rd level engineering diagrams (E1) • Prepares unit test plans for facility / hardware items (E2) • Prepares the systems integration strategy (E3) • Provides technical impact assessments to requirements and high level design changes (M1)
Data Management	• Prepares Physical Data Model (E1) • Prepares data management test plans including access, integrity, replication functions, etc. (E2) • Provides input to integration strategy (E3) • Provides input to support materials (E4)
Software Engineering	• Prepares detail software design specifications(E1) • Prepares software presentation and processing unit / component test plans (E2) • Provides input to integration strategy (E3) • Provides input to support materials (E4)
CM	• Control work product revisions i.e., manages CM library (M1) • Documents change history (M1)
QA	• Assesses and approves adequacy of work products prior to Critical Design Review (M3) • Ensures completion of CDR checklist (M3)
Test and Evaluation	• Assists in preparation of unit / component test plans (E2) • Design test procedure elements (logs, tracking) (E2) • Ensures test plans relate to requirements e.g., data range, response rates, etc. (E2) • Prepares user and operator support material outlines (E4) • Identifies applicable elements of CDR checklist (M3)
Operations and Maintenance	• Defines operational support procedures (E1)

6.3 Critical Design Review Checklist

Purpose and Objectives:

The CDR milestone provides a final checkpoint of the system and software design before actual construction begins. Designs are evaluated for logical coherency and completeness. Developer test plans are evaluated to determine whether they provide sufficient coverage of each system unit or component to be implemented.

Objectives for the CDR include

- Establish confidence in the detailed design to meet system requirements
- Establish confidence in the test plans that the system will operate with the expected level of reliability
- Provide a basis for concurrence of the project to move to the next activity

Because reviewing detailed designs can be tedious to a large audience, the CDR will not be a forum for discussing a specific routine's logical sequence or configuration. Rather, general descriptions of the detailed design will be provided through diagrams and discussion. If the system is large in scope, then it may be worthwhile to apply an 80/20 rule such that the CDR focuses on the 20% of items that will encompass 80% of system usage. Detailed designs should be vetted through peer reviews or other mechanisms prior to the CDR. However, the CDR may generate questions as to why a unit was designed a particular way. The development team should be prepared to respond to those questions and ensure all system stakeholders have a good understanding of how the system will be built.

Presenters

- Development team

Participants

- Product Owner(s)
- Guidance Team
- QA
- Operations and Maintenance

When:

This review is conducted after tasks for the Detailed Design activity are complete and QA has confirmed the products are ready for review. Multiple sessions may be conducted depending on the size and complexity of the project. When multiple sessions are used it is recommended that interim reviews be conducted against major system components with a final review to evaluate integration aspects.

Key Issues

- Have individual units been specified in accordance with Software and Systems Engineering guidelines?
- Will designs facilitate testing per the Test Strategy?
- Does the design appear to meet all functional and performance requirements?
- Have changes to requirements and high-level design baselines been recognized and assessed for their impact to success of the project?

Entry Criteria

The following items should be done prior to conducting the review

- Design materials are distributed 2 weeks prior to review

Exit Criteria

The review process is not considered closed until the following items are complete.

- All issues raised during the review have a documented resolution
- The Product Owner agrees to the Detailed Design
- The Guidance Team approves of schedule and resource revisions
- QA confirms completion of the Review Checklist

Presentation of Detailed Design

Overview of design strategies

Assumptions used in design

Infrastructure Architecture
- Impacts of detailed design on baseline requirements and high-level design
- 3rd level engineering diagrams
- Hardware / Network / Facility Units
- Unit descriptions
  - Assumptions
  - Description and implementation overview

Interface Design
- Descriptions
- Revisions to MOUs, Data Sharing agreements, ICDs

Software Units
- Impacts of detailed design on baseline requirements and high-level design
- Software Unit Descriptions
  - Design assumptions
  - Design diagrams / overview
- COTS Implementation Design
  - Assumptions
  - Overview / Description
  - Justification for approach
- Physical Data Model
  - Overview of differences from LDM & Data Model Implementation Strategy

System Integration Plan
- Delivery procedures
- CI build procedures
- Confirmation procedures

Software Design Checklist

General

- Are assumptions used in the design valid
- Is error handling and recovery specified in accordance with high-level design

Custom Development

- Does the design meet the standards as specified in the Software Development Plan (such as complexity limitations or specification method)?
- Can all requirements be traced to detailed design units?
- Have all interfaces been reconciled (e.g., do parameters of a function - or method - match the call to that function)?
- Does the detailed design specify recovery actions for failed logical transactions (such as a rollback on failure)?
- Does the detailed design support performance and security requirements?

COTS

- Have configuration specifications been given supporting justification?
- Does the configuration support performance and security requirements?

Hardware / Network / Facilities Checklist

- Have user access profiles been specified according to security constraints?
- Does the design specify how access will be controlled and monitored?
- Does each unit have a defined configuration?
- Does each unit's configuration support performance requirements?
- Does the design minimize the need for operational monitoring?
- Have system failure / disaster recovery procedures been defined in accordance with the high-level design?
- Does the design support all connectivity requirements with external systems?
- Does the design establish how internal and external environments will be separated if needed (e.g., Internet vs. intranet requirements)
- Does the design specify how system components and the operating environment will be monitored?
- Have all software units been allocated to hardware components?

System Integration Plan

- Does the plan have defined steps for building and baselining Configuration Items?

Project Management Presentation

Summary report of requirements changes
Impacts to Schedules / Resources
Risk Mitigation efforts
Schedule for future activities

Project Management Checklist

- Are actual vs. estimated items on target? If not, have corrective actions for schedule slippage or cost overruns been implemented?
- Is the schedule for future work realistic?
- Have project risk factors been actively monitored?
- Have risk mitigation efforts been implemented where required?

Proposed Schedule / Resource Revisions

- Have reasons for the schedule or resource changes been adequately tracked to the cause and documented?
- Do project revisions increase the risk category assigned to the project? If so, has the project taken appropriate steps to control the activities and implement the rigor necessary?
- Can the product mission be achieved with the proposed changes?

6.4 Detailed Design Product Templates

DETAILED SYSTEM DESIGN TEMPLATE		
Identification Number	**Title**	**Purpose**
NARA-IT-T-0007	Detailed System Design Document	The Detailed System Design Document describes the design of the system at the component-level. The architecture design, interface design, software design, and database design are included.
Content Description		
The contents of a Detailed System Design Document should include the following topics:		
Introduction. The introduction sets the context of the document by identifying the Product Plan, Project Plan, and system to which it pertains, the background of the project, the scope of the development effort, an overview of the system, operational policies and constraints, document overview, relationship to other documents, and referenced documents.		
Approach. Describe the approach or approaches used in the design. If a particular technique or diagramming method is used, briefly define terminology and notations.		
Assumptions. Identify any assumptions being used in creating the detailed design. Risks associated with design assumptions should be highlighted as to their potential impact and mitigation.		
System Component Architecture Design. This topic area should include the same information as the comparable section in the Preliminary System Design Document only at the detailed system component level. Level 3 Systems Engineering Diagrams must be used to describe the architecture. See Appendix D for a definition and specific requirements pertaining to the SED3.		

DETAILED SYSTEM DESIGN TEMPLATE

Identification Number	Title	Purpose
NARA-IT-T-0007	Detailed System Design Document	The Detailed System Design Document describes the design of the system at the component-level. The architecture design, interface design, software design, and database design are included.

Content Description

Interface Component Architecture Design.
This topic should describe the design for the interfaces with external systems at the component level. A Systems Engineering Level 3 Diagram should be used to depict the architecture interface specifications. A Physical Data Model (PDM) should be used to describe the data interfaces where applicable. See Appendix F for a definition and specific requirements pertaining to the PDM. See Appendix G for data modeling naming standards and conventions. Interface descriptions should include the standard data element name, technical name, abbreviation or synonymous names, data types, size and format, units of measurement, range or enumeration of possible values, accuracy and precision, priority, timing, frequency, volume, sequencing, and other constraints, security and privacy constraints, and sources and recipients.

In addition, interface descriptions should include the characteristics of data element assemblies (records, messages, files, arrays, displays, reports, etc.) that the interfacing entities will provide, store, send, access, receive, etc such as the identifier, data elements in the assembly and their structure, medium, visual and auditory characteristics of displays and other outputs, relationships among assemblies such as sorting/access characteristics, priority, timing, frequency, volume, sequencing, and other constraints, such as whether the assembly may be updated and whether business rules apply, security and privacy considerations, and sources and recipients.

Descriptions of communications methods should include communication links/bands/frequencies/media and their characteristics, message formatting, flow control, data transfer rate, routing, addressing, and naming conventions, transmission services, and security and privacy such as encryption, user authentication, compartmentalization, and auditing.

Protocol descriptions should include priority/layer of protocol; packeting, including fragmentation and reassembly, routing, and addressing; legality checks, error control, and recovery procedures; synchronization, including connection establishment, maintenance, termination; status, identification, and any other reporting features.

Descriptions should include other characteristics such as physical compatibility of the interfacing entities (dimensions, tolerances, loads, voltages, plug compatibility, etc.)

Software Component Design.
This topic should describe the components of a specific software application. It should include design decisions regarding inputs each component will accept and outputs it will produce, including interfaces with other systems components. It should include design decisions on component behavior in response to each input or condition, including the actions the component will perform, response times and other performance characteristics, description of the physical systems modeled, selected equations/algorithms/rules, and handling of unallowed inputs or conditions.

Software Design should be documented in accordance with the methodology defined in the project's Software Development Plan.

DETAILED SYSTEM DESIGN TEMPLATE		
Identification Number	**Title**	**Purpose**
NARA-IT-T-0007	Detailed System Design Document	The Detailed System Design Document describes the design of the system at the component-level. The architecture design, interface design, software design, and database design are included.
Content Description		

Detailed Database Design.

Detailed Design should include the system's Physical Data Model (PDM) developed in accordance with NARA Data Deliverable guidelines.

Physical Data Model includes

- Documentation of table and column designs, domain definitions, and integrity controls.
- Documentation of DBMS configuration and technical management parameters such as sizing, indices, access views, locking, tuning, backup, recovery, buffering, etc.
- Source data definition language statements to be used by the DBMS software to generate the actual physical database structures, referential integrity, and triggers/edits.

DEVELOPER TEST PLAN TEMPLATE		
Identification Number	**Title**	**Purpose**
NARA-IT-T-0008	Developer Test Plan	The Developer Test Plan describes the unit and component testing prior to delivery of packages to the CM library.
Content Description		
The contents of a Developer Test Plan should include the following topics:		
Introduction. The introduction sets the context of the document by identifying the Product Plan, Project Plan, and system to which it pertains, the background of the project, the scope of the development effort, an overview of the system, operational policies and constraints, document overview, relationship to other documents, and referenced documents.		
Test Items. This topic should identify the test items including their version/revision level. Also to be specified are the characteristics of the transmittal media which impact hardware requirements or indicate the need for logical or physical transformations before testing begins (such as programs transferred from tape to disk). References to the applicable documentation including the Product and Project Plans, Requirements Document, Preliminary and Detailed Design Documents, Users Manual, Operations Manual, Installation and Transition Plan, etc. should also be made.		
Features to be Tested/Not Tested. This topic should identify all software features and combinations of features to be tested. The test design specification associated with each feature should also be specified. The Pass/Fail criteria for each test item should be documented. Features that will not be tested should be identified along with the rationale for not testing them.		
Approach. This topic should describe the overall testing approach. The approach should be described in sufficient detail to permit identification of the major tasks and the estimation of the time to do each one. Included in the description should be the minimum degree of comprehensiveness desired, the techniques used to judge the comprehensiveness of the testing effort (such as determining which statements have been executed at least once), additional completion criteria (such as error frequency), the techniques to be used to trace results to requirements, and significant constraints such as test item availability, resource availability, and deadlines. All developer test plans should be written in adherence to the project's Test Strategy. The approach should indicate how it is supporting the defined strategy.		
Test Deliverables. Deliverables should be specified including the Test Description Document with test design specifications, test case and procedure specifications, etc. Test input and output data should be identified as deliverables. Test tools should also be considered.		
Test Tasks. This topic should include the set of tasks necessary to prepare for and perform testing. Intertask dependencies and special skills required should also be specified.		
Responsibilities. This topic area should identify the groups responsible for managing, designing, preparing, executing, witnessing, checking, and resolving test events. Responsibilities for providing test items, test data, and the test environment (hardware, software, tools, facilities, licenses, etc.) should also be specified.		
Schedule. This topic should include the testing schedule.		

7 Development

7.1 Purpose

Development tasks constitute the actual construction of the phase deliverables. Depending on the type of system being built this may consist of system integration activities for COTS components, complete coding efforts, or a combination of the two. Unit and component testing is conducted after construction to ensure a *reasonable* likelihood that the system will work as intended. Time should be allocated to unit testing and error correction that correlates with the risk of unit failure. It is not unusual for the testing activity to exceed the coding or configuration time. Development of the end-to-end system test plan is included in this activity so that feedback from the developer tests may be incorporated into the system level evaluation (e.g., areas that proved troublesome during developer testing).

Management again controls changes to requirements and design. Results of developer testing may be used to assess system stability and can be a good indicator of the likelihood that the system will fulfill its objectives. A high level of errors during developer testing may point to the need for reviewing the design of those components or areas for closer examination. On a longer term basis these factors may point to areas for increased training or process improvements.

The Test Readiness Review is conducted to ensure that the development team is prepared to perform the systems integration and conduct end-to-end system testing. Since actual code and test results are not reviewed during the TRR, inspections may be held by the Product Owner and Guidance Team during the development process to ensure that activities are being conducted in accordance with the project plan. Note that Configuration Items are constructed during this phase, but will not actually be a specified deliverable until after Acceptance Testing is performed successfully.

7.2 Development Task Outline

The following timeline figure shows the typical sequence of Development tasks and work product creation.

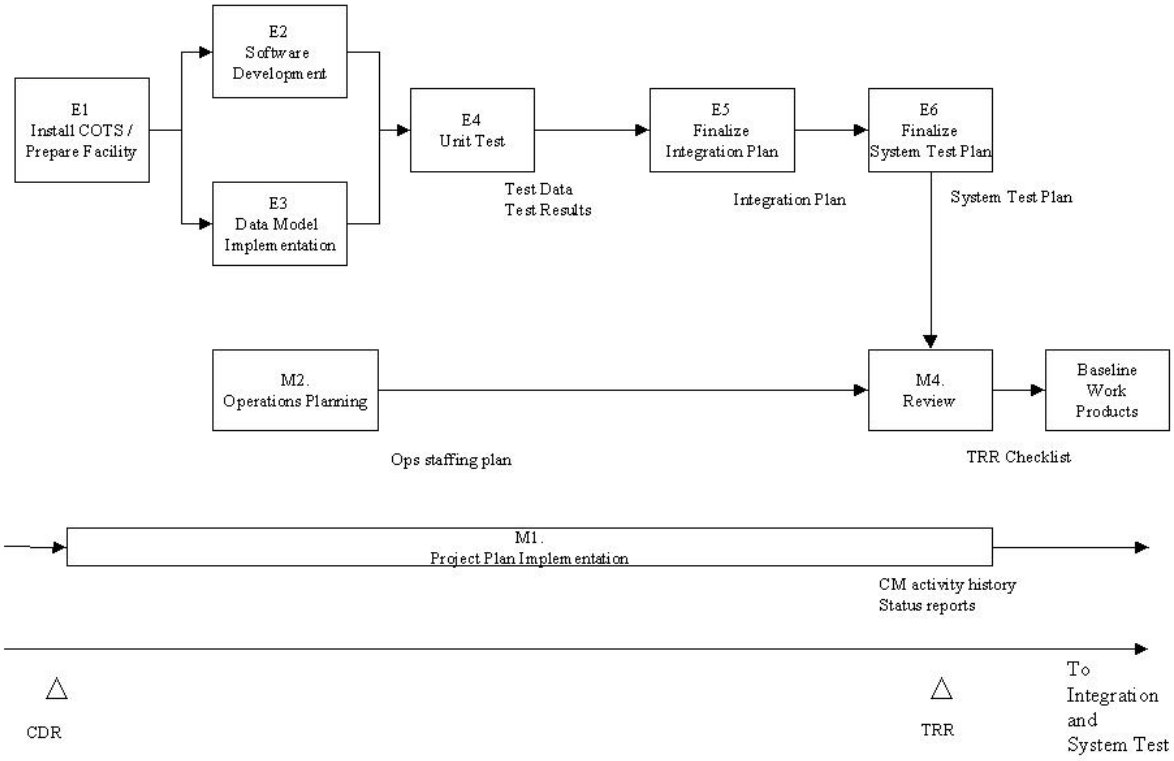

Figure 7-1 Development Activity Timeline

Table 7-1 Development Performance Requirements

Entry Criteria	**Development Entry Criteria**
Products and activities to be completed before beginning a new software development activity.	System Test Plan approved by Product Owner and Guidance Team Resource allocation changes approved by Guidance Team Detail schedule prepared for Development activity
Work Products	**Development Work Products**
Products and activities to be completed during the current software development activity.	Developer Test Results Final Integration Plan Final System Test and Evaluation Plan Security Integration and Acceptance Security Control Integration Security Certification & Accreditation (C&A) Completed Draft Records Schedule
Exit Criteria	**Development Exit Criteria**
What must be finished before an ongoing activity is designated complete	Integration and System Test Plans approved by Product Owner and Guidance Team Issues from Test Readiness Review resolved and documented Detail Schedule for Integration and System Test tasks Integration and System Test Plans baseline established Development conforms to the Software Architecture from the EA and complies with NARA's IT standards.

Engineering Tasks

E1. Install COTS hardware and software
   a. Prepare facility for installation
   b. Perform acceptance testing of COTS in accordance with supplier agreements
   c. Install and configure items according to design

E2. Perform custom software coding
   a. Customize or code functional units per design
   b. Perform walkthroughs, peer reviews, other checks in accordance with QA plan

E3. Implement the Physical Data Model
   a. Create tables and other database objects
   b. Establish testing environments

E4. Perform Unit / Component Testing
   a. Create test data
   b. Perform testing in accordance with test plans
   c. Log results
   d. Document and fix errors
   e. Prepare results report

E5. Finalize Integration Plan
   a. Define CI build procedures for developed items

E6. Finalize System Test Plan
   a. Design system tests in accordance with project Test Plan
   b. Coordinate interface testing with external systems

Management Tasks

M1. Implement project plan elements for Development activity
   a. Control changes to requirements / design through CCB
   b. Monitor schedule / Prepare detail for next activity
   c. Perform risk management
   d. Generate status reports

M2. Draft the Operations Management Plan
   a. Assess operation procedure definitions for operational staffing needs
   b. Prepare operations staffing and training plan

M3. Perform Test Readiness Review (TRR)
   a. Prepare and Distribute TRR materials (checklist, work products)
   b. Review materials prior to meeting
   c. Conduct TRR meeting
   d. Resolve open issues
   e. Establish configuration baselines for work products

Roles	Responsibilities
Product Owner	• Approves requirements and design changes (M1) • Approves work products during TRR (M4)
Guidance Team	• Ensures conformance to NARA standards (M4) • Approves schedule / budget changes and continuation to next activity (M1 / M4)
Project Management	• Facilitates CCB (M1) • Prepares schedule / budget impact assessments for requirements and design changes (M1) • Conducts TRR (M4) • Completes TRR checklist (M4)
Systems Engineering	• Confirms receipt of COTS items and completion of acceptance tests (E1) • Performs facility and hardware integration (E1) • Installs and configures COTS hardware components (E1) • Conducts facility / hardware component testing (E4) • Prepares technical impact assessments for requirements and design changes (M1) • Provides input to system test plans (E6)
Data Management	• Configures COTS database management components (E1) • Implements physical data models (E3 / E4) • Prepares test databases (E3) • Conducts unit testing for data management components (E4)
Software Engineering	• Configures COTS software components for presentation and processing layers (E1) • Develops or customizes software (E2) • Conducts software unit / component testing (E4)
CM	• Controls work product revisions i.e., manages CM library (M1) • Documents change history (M1)
QA	• Reviews development units for completeness (E5) • Assesses and approves adequacy of work products prior to Test Readiness Review (M4) • Ensures completion of TRR checklist (M4)
Test and Evaluation	• Prepares system test plans (E6) • Identifies applicable elements of TRR checklist (M4)
Operations and Maintenance	• Identifies system operations resource needs / training requirements (M2)

7.3 Test Readiness Review Checklist

Purpose and Objectives:

The TRR evaluates the work products from system construction tasks to determine whether the built system is ready to conduct end-to-end testing. Objectives for the TRR include

- Demonstrate that the system has been constructed per the design
- Demonstrate that sufficient time was spent on developer testing and error correction
- Ensure that a system test plan has been prepared that will adequately exercise the system and validate it against the requirements

The TRR should provide a checkpoint to evaluate those areas that may present the highest risk to the system test and operation. Components that had high error rates during unit testing or that initiated a number of requirements or design changes are the types of indicators that should be assessed during the TRR for determining system stability and movement to the next activity.

Presenters

- Development Team

Participants

- Product Owner(s)
- Guidance Team
- QA

When:

The TRR is conducted when all configuration items have been built, tested, and delivered. Tasks defined for the Development activity up to the review and baseline should be complete.

Key Issues

- Have all system components been constructed in accordance with design?
- Has sufficient testing been done to achieve the needed level of system stability?
- Do project indicators suggest that the project is ready to begin system testing?

Entry Criteria

The following items should be done prior to conducting the review

Distribution of the System Test Plan 2 weeks prior to review
Distribution of Test and Integration results 1 week prior to review

Exit Criteria

The review process is not considered closed until the following items are complete.

- All issues raised during the review have a documented resolution
- The Product Owner approves of the System Test Plan
- The Guidance Team approves of schedule and resource revisions
- QA confirms completion of the Review Checklist

Presentation of Development Efforts

Summary of Changes to Requirements / Design baselines
Developer Unit / Component Testing Results
Integration Plan
System Test Plan

Product Checklist

Developer Testing Results

- Have test errors been documented with a resolution?
- Was testing conducted in accordance with the Test Strategy?
- Have items or areas that experienced high failure rates been given sufficient review?
- Have individual units and components been evaluated for performance?

System Test Plan

- Is the plan written in accordance with the approach established in the Project Plan?
- Have test (use) cases been defined to check all critical system level requirements?
- Does the system test include the cooperation of external systems?
- Are appropriate resources identified to conduct testing?
- Has data been prepared to conduct the system test?

System Integration Plan

- Does the plan have defined steps for building and baselining Configuration Items?

Project Management Presentation

Summary report of requirements changes
Impacts to Schedules / Resources
Current project status against schedule and budget
Risk Mitigation efforts
Schedule for future activities

Project Management Checklist

- Are actuals vs. estimated items on target? If not, have corrective actions for schedule slippage and cost overruns been implemented?
- Is the schedule for future work realistic?
- Have project risk factors been actively monitored?
- Have risk mitigation efforts been implemented where required?

Proposed Schedule / Resource Revisions

- Have reasons for the schedule or resource changes been adequately tracked to the cause and documented?
- Do project revisions increase the risk category assigned to the project? If so, has the project taken appropriate steps to control the activities and implement the rigor necessary?
- Can the product mission be achieved with the proposed changes?

7.4 Development Product Templates

DEVELOPER TEST REPORT TEMPLATE		
Identification Number	**Title**	**Purpose**
NARA-IT-T-0010	Developer Test Report	The Developer Test Report describes the results of testing individual units and components.
Content Description		
The contents of a Developer Test Report should include the following topics:		
Introduction. The introduction sets the context of the document by identifying the Product Plan, Project Plan, and system to which it pertains, the background of the project, the scope of the development effort, an overview of the system, operational policies and constraints, document overview, relationship to other documents, and referenced documents.		
Test Summary Report. This topic area should summarize the results of the designated testing activities and provide evaluations based upon the results. Analysis of the results should be done according to the project's Test Strategy.		
Test Incident Report. This topic area should document any event that occurs during testing which requires investigation.		
Test Log. This topic area should include a chronological record of relevant details about the execution of tests.		

SYSTEM TEST PLAN TEMPLATE		
Identification Number	**Title**	**Purpose**
NARA-IT-T-0009	System Test Plan	The System Test Description describes the detailed technical tests to be performed after delivery to the CM library and all Configuration Items are built.

Content Description

The contents of the System Test Plan should include the following topics:

Introduction.
The introduction sets the context of the document by identifying the Product Plan, Project Plan, and system to which it pertains, the background of the project, the scope of the development effort, an overview of the system, operational policies and constraints, document overview, relationship to other documents, and referenced documents.

Indicate how the System Test Plan supports the project's Test Strategy.

Test-Design Specification.
A test design specification includes a detailed description of the following:

- Features to be tested
- Approach refinements
- Test identification
- Feature pass/fail criteria

If these items are described in adequate detail within the Developer Test Plan, then the plan should be referenced in place of this topic.

Test-Case Specification.
This topic should specify inputs, predicted results, and a set of execution conditions for a test item. Specific topics should include test items, input and output specifications, environmental needs, special procedural requirements, and intercase dependencies (order of test case execution).

Test Procedure Specification.
This topic area should specify the steps for executing a set of test cases, such as setup, startup, measurement, shut down, restart, stop, and wrap up steps.

INTEGRATION PLAN TEMPLATE		
Identification Number	**Title**	**Purpose**
NARA-IT-T-0019	Integration Plan	Provides details on how system components are to be delivered to the CM manager and integrated to provide the full system or sub-system implementation.
Content Description		
The contents of an Integration Plan should include the following topics:		
Introduction. The introduction sets the context of the document by identifying the Product Plan, Project Plan, and system to which it pertains, the background of the project, the scope of the development effort, an overview of the system, operational policies and constraints, document overview, relationship to other documents, and referenced documents.		
Delivery Procedures. Specify how all system items will be delivered to the CM library. This includes any issues with tagging items with Release Identifications, sign-offs with workflow control, and anything else necessary to ensure that a baseline of all deliverables can be created.		
Integration Procedures. This should identify the tasks needed to integrate CIs into a complete system. In some instances this can include listing the order that CIs must be implemented and any special instructions for ensuring they will work together as planned.		
Problem Resolution Procedures. Identify what should occur if there are any problems encountered during the delivery or integration of system components. This section should also indicate the responsible parties for integration tasks.		

8 Integration and System Test

8.1 Purpose

The first goal for this activity is a successful build of defined Configuration Items in accordance with the systems integration plan. The integration activity ensures that all system components are tagged with the release identification and establishes an environment to conduct system testing. The system test evaluates operation of the system as a single entity performed in a manner that emulates production.

Management during this activity focuses on ensuring errors are addressed and preparing for the delivery and deployment of the system. Changes to requirements or design should be cautiously evaluated. The Product Owner, Guidance Team, and Project Manager may need to negotiate how to respond to requirements that have not been met. Requirements that have not been met may have to be dropped or delayed until the next system release. Significant system changes are discouraged at this point unless they are seen as critical to the success of the project phase. If that is the case, then the project may need to revert to an earlier activity so that the impact of changes can be managed accordingly.

8.2 Integration and System Test Task Outline

The following timeline figure shows the typical sequence of Integration and System Test tasks and work product creation.

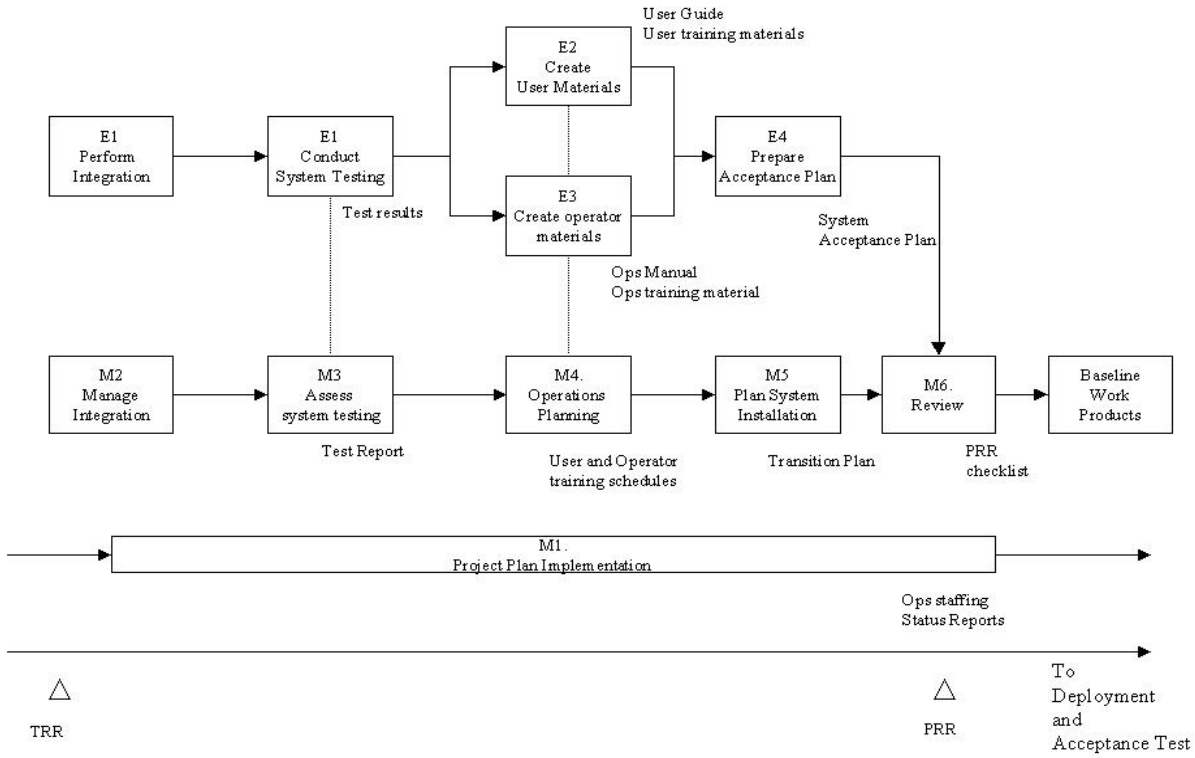

Figure 8-1 Integration and System Test Timeline

Table 8-1 Integration and System Test Performance Requirements

Entry Criteria	Integration and System Test Entry Criteria
Products and activities to be completed before beginning a new software development activity.	Integration and System Test Plans approved by Product Owner and Guidance Team Issues from Test Readiness Review resolved and documented Detail Schedule for Integration and System Test tasks Integration and System Test Plans baselined
Work Products	**Integration and System Test Work Products**
Products and activities to be completed during the current software development activity.	System Test Report Version Description Document Installation and Transition Plan Acceptance Test Plan User Guides Operations Manual Training Plans and Materials
Exit Criteria	**Integration and System Test Exit Criteria**
What must be finished before an ongoing activity is designated complete	System Test completed successfully Acceptance Test Plan approved by Product Owner User and Operator materials prepared and approved by Product Owner and Operations System transition and installation plan approved by Product Owner and Operations Resource allocation changes approved by Guidance Team Issues from Production Readiness Review resolved and documented Detail schedule prepared for Acceptance Testing tasks Work products baselined

Engineering Tasks

E1. Implement Integration strategy
   a. Deliver software packages in accordance with Integration Plan
   b. Perform system Configuration Item builds
   c. Prepare Version Description Document

E2. Conduct System Testing
   a. Prepare test data
   b. Run defined tests
   c. Log test results
   d. Resolve Test Incidents
   e. Integrate patches into system baselines
   f. Compare the collected test, inspection, or review results with established evaluation criteria to assess the degree of success and prepare System Test report.

E3. Update and finalize User Materials
   a. Update draft User Guide
   b. Create user training materials

E4. Update and finalize Operator Materials
   a. Update draft Operations Manual
   b. Create operator training materials

E5. Prepare System Acceptance test plans
   a. Define acceptance criteria
   b. Define test cases

Management Tasks

M1. Implement project plan elements for Integration and Test activity
   a. Control changes to requirements / design through CCB
   b. Monitor schedule / Prepare detail for next activity
   c. Perform risk management
   d. Generate status reports

M2. Manage system integration
   a. Verify the receipt of each system element required to assemble the system in accordance with the physical architecture.
   b. Ensures unit testing and reviews have been performed in accordance with QA procedures

M3. Create System Test report
   a. Address risks associated with level of testing performed
   b. Prepare justification for operational readiness

M4. Operations Planning
   a. Initiate Operations staffing plan

M5. Prepare System Installation and Transition plan
   a. Define how deployment of Configuration Items will take place
   b. Address migration and database population issues
   c. Define the development freeze period

M6. Perform Production Readiness Review (PRR)
   a. Prepare and Distribute PRR materials (checklist, work products)
   b. Review material prior to meeting
   c. Conduct PRR meeting
   d. Resolve open issues
   e. Establish configuration baselines for work products

Roles	Responsibilities
Product Owner	• (May) participate in systems testing (E1) • Develops ops staffing plan (M4) • Approves system acceptance test plan (M5) • Approves transition plan (M5)
Guidance Team	• Approves schedule / resource changes and continuation to next activity (M1 / M5) • Ensures adherence to NARA standards (M5)
Project Management	• Completes Integration checklist (M3) • Implements project plan Integration and Test phase elements (M1) • Prepares operational readiness justifications (M3) • Prepares transition and installation plan(M5) • Conducts PRR (M6) • Completes PRR checklist (M6)
Systems Engineering	• Resolves system configuration Test Incident reports (E1) • Provides input to User Guide (E2) • Provides input to Operations Manual (E3) • Provides input to Acceptance Test plan (E4) • Provides input to operations readiness justification (M3) • Provides input to transition plan (M4)
Data Management	• Resolves data management Test Incident reports (E1) • Provides input to User Guide (E2) • Provides input to Operations Manual (E3)
Software Engineering	• Resolves software Test Incident reports (E1) • Provides input to User Guide (E2) • Provides input to Operations Manual (E3)
CM	• Performs Configuration Item builds (E5) • Creates Version Description Document (E5) • Integrates system fixes into build (E1) • Documents change history (M1)
QA	• Ensures completeness of integration elements in accordance with integration strategy (E5) • Assesses and approves adequacy of work products prior to Production Readiness Review (M5) • Ensures completion of PRR checklist (M5)
Test and Evaluation	• Performs System Testing (E1) • Prepares System Test report (E1 / M3) • Identifies applicable elements of PRR checklist (M5) • Prepares User and Operator Materials (M5)
Operations and Maintenance	• Identifies Operations staff (M4)

8.3 Production Readiness Review Checklist

Purpose and Objectives:

The PRR process evaluates the results of the end-to-end system testing and preparations for the new system release. In particular, user guides, operating manuals, and various training materials are reviewed to determine whether the system can be implemented with success.

Objectives for the PRR include

- Demonstrate the system performs as intended and with expected reliability
- Ensure materials for users and operators will provide an appropriate level of support
- Show that migration and deployment activities have been planned to take place with minimal disruption
- Develop consensus among stakeholders that the system can be implemented with successful results

Issues that were unresolved from system testing must have been negotiated for removal from the current build and should be highlighted during this review.

Presenters

Project Management
Systems Engineering
Test and Evaluation

Participants

Product Owner(s)
Guidance Team
Appropriate development staff
QA

When:

After system testing is complete and all materials are ready for production.

Key Issues

- Has system implementation been planned so that users can easily accept the changes?
- Can operational use be supported by system test results?
- Are supporting materials acceptable?
- Have all components been delivered, integrated, and baselined in accordance with the System integration plan?

Entry Criteria

The following items should be done prior to conducting the review

- Distribution of Acceptance Test Plan, Installation and Transition Plan, User materials, and Operator materials 2 weeks prior to review
- Distribution of System Test Results 1 week prior to review

Exit Criteria

The review process is not considered closed until the following items are complete.

- All issues raised during the review have a documented resolution
- System readiness accepted by Product Owner and Guidance Team
- QA confirms completion of the Review Checklist

Presentation of Products

Version Description Document / Integration Results
System Test Results
Acceptance Test Plan
User Materials
Operator Materials

Complete the following Product Checklists

Version Description Document

- Has the software been built in accordance with the Integration Plan?
- Does the Version Description Document cover all the expected phase deliverables?
- Does the Version Description Document contain all build dependencies?
- Have hardware configuration items been implemented in accordance with the Integration plan?
- Have configuration files or parameter settings been established in accordance with the Integration Plan

System Testing

- Have test errors been documented with a resolution?
- Was each test case conducted in accordance with the project's test plan?
- Have items or areas that experienced high failure rates been given sufficient review?

Acceptance Test Plan

- Are criteria defined that will determine whether the system can fulfill its mission?
- Are test cases defined that will cover all acceptance criteria?
- Has the test plan been developed in accordance with the project Test Strategy?
- Is testing sufficient according to the risk level of the project?

User Support Materials

- Are instructions provided in sufficient detail to cover all critical processes?
- Are instructions provided to help users recover from errors?
- Are instructions clear and unambiguous?
- Do the training plans provide methods for assessing user understanding?

Operator Support Materials

- Are instructions provided on how to monitor the system operation?
- Are instructions provided in sufficient detail to perform all routine procedures?
- Are instructions provided to help operators respond to errors and system failures?
- Are instructions clear and unambiguous?
- Do the training plans provide methods for assessing operator understanding?

Project Management Presentation

Review of changes to Requirements and Design
Current project status against schedule and budget
Operational Readiness Report
Transition and Installation Plan
Schedule Review
Risk Mitigation efforts

Transition and Installation Plan Checklist

- Are all materials needed for transition to the new system specified?
- Is the change over to the new system planned with minimum disruption to current environment?
- Are migration efforts from existing systems defined?
- Has coordination with external systems been defined?

Project Management Checklist

- Are actuals vs. estimated items on target? If not, why? Have mitigation actions been implemented?
- Is the schedule for future work realistic?
- Have project risk factors been actively monitored?
- Have risk mitigation efforts been implemented where required?

8.4 Integration and System Test Product Templates

INSTALLATION AND TRANSITION TEMPLATE		
Identification Number	**Title**	**Purpose**
NARA-IT-T-0012	Installation and Transition Plan	The Installation and Transition Plan identifies the hardware, software, and other resources needed for lifecycle support of deliverable software and describes the developer's plans for transitioning deliverable items to the Project. It also includes plans for installing hardware and software at the user sites, including preparations and conversion from existing systems.
Content Description		
The contents of a Installation and Transition Plan should include the following topics:		
Introduction. The introduction sets the context of the document by identifying the Product Plan, Project Plan, and system to which it pertains, the background of the project, the scope of the development effort, an overview of the system, operational policies and constraints, document overview, relationship to other documents, and referenced documents.		
Approach. The general approach for deploying the system should be specified under this topic. The approach is likely to be either 1) immediate cutover, 2) parallel operations, or 3) phased initiation. Parallel and phased deployments are expected to be more likely when replacing existing systems or when operating in multiple sites. Justification for the selected approach should be included.		
Support Resources. This topic should describe the resources needed to perform installation and transition procedures. Resources include facilities, hardware, software, personnel, and documentation.		
Tasks. This topic should describe the tasks required to prepare for installation and transition including planning, coordination, and preparation for installation; ensuring that all manuals are available when needed, planning and conducting training, providing computer support and technical assistance for the installation, and providing for conversion from the current system.		
Schedule. This topic should include the schedule for installation and transition activities and milestones.		
Installation and Transition Procedures. This topic should describe the specific procedures for installation and transition such as installing software, checking out the software once it is installed, initializing databases and other software with site-specific data, conversion from the current system, and dry run of the procedures in operations and users manuals.		

ACCEPTANCE TEST PLAN TEMPLATE		
Identification Number	**Title**	**Purpose**
NARA-IT-T-0011	Acceptance Test Plan	The Acceptance Test Plan describes the process that the built and delivered system should undergo to gain Product Owner acceptance of the defined release and begin Production.

Content Description

The contents of an Acceptance Test Plan should include the following topics:

Introduction.
The introduction sets the context of the document by identifying the Product Plan, Project Plan, and system to which it pertains, the background of the project, the scope of the development effort, an overview of the system, operational policies and constraints, document overview, relationship to other documents, and referenced documents.

Test Items.
This topic should identify the test items including their version/revision level. Also to be specified are the characteristics of the transmittal media which impact hardware requirements or indicate the need for logical or physical transformations before testing begins (such as programs transferred from tape to disk). References to the applicable documentation including the Product and Project Plans, Requirements Document, Preliminary and Detailed Design Documents, Users Manual, Operations Manual, Installation and Cutover Plan, Test and Evaluation Plan, etc. should also be made.

Features to be Tested/Not Tested.
This topic should identify all software features and combinations of features to be tested. The Pass/Fail criteria for each test item should be documented. Suspension and resumption requirements should also be described. The test design specification associated with each feature should also be specified. The features which will not be tested should be identified along with the rationale for not testing them.

Approach.
This topic should describe the overall testing approach. The approach should be described in sufficient detail to permit identification of the major tasks and the estimation of the time to do each one. Included in the description should be the minimum degree of comprehensiveness desired, the techniques used to judge the comprehensiveness of the testing effort (such as determining which statements have been executed at least once), additional completion criteria (such as error frequency), the techniques to be used to trace results to requirements, and significant constraints such as test item availability, resource availability, and deadlines.

Acceptance Criteria.
This topic area should include the minimum requirements for acceptance of the system (i.e., the tolerance for remaining test errors and the means for fixing them before acceptance can take place).

Test Deliverables.
Deliverables should be specified including the Developer Test Plan and Test Description Document with test design specifications, test case and procedure specifications, etc. Test input and output data should be identified as deliverables. Test tools should also be considered.

Test Tasks.
This topic should include the set of tasks necessary to prepare for and perform testing. Intertask dependencies and special skills required should also be specified.

Responsibilities.
This topic area should identify the groups responsible for managing, designing, preparing, executing, witnessing, checking, and resolving test events. Responsibilities for providing test items, test data, and the test environment (hardware, software, tools, facilities, licenses, etc.) should also be specified.

Schedule.
This topic should include the testing schedule.

SYSTEM TEST REPORT TEMPLATE		
Identification Number	**Title**	**Purpose**
NARA-IT-T-0010	System Test Report	The System Test Report describes the results of the system test after building of Configuration Items.
Content Description		
The contents of a System Test Report should include the following topics:		
Introduction. The introduction sets the context of the document by identifying the Product Plan, Project Plan, and system to which it pertains, the background of the project, the scope of the development effort, an overview of the system, operational policies and constraints, document overview, relationship to other documents, and referenced documents.		
Test Summary Report. This topic area should summarize the results of the designated testing activities and provide evaluations based upon the results. Analysis of the results should be done according to the project's Test Strategy.		
Test Incident Report. This topic area should document any event that occurs during testing which requires investigation.		
Test Log. This topic area should include a chronological record of relevant details about the execution of tests.		

TRAINING PLAN TEMPLATE		
Identification Number	**Title**	**Purpose**
NARA-IT-T-0013	Training Plan	The Training Plan describes training activities, responsibilities, and schedules for training the operations staff and users in the operation and maintenance of the system.
Content Description		
The contents of a Training Plan should include the following topics:		
Introduction. The introduction sets the context of the document by identifying the Product Plan, Project Plan, and system to which it pertains, the background of the project, the scope of the development effort, an overview of the system, operational policies and constraints, document overview, relationship to other documents, and referenced documents.		
Tasks. This topic area should describe the tasks for preparing, executing, and managing training for end-users and/or operational support staff for the specified system.		
Responsibilities. This topic should specify the groups and individuals responsible for preparing, executing, and managing the specific training programs associated with the specific system.		
Training Material Description. This topic area should describe the required training material including curriculum, media, and format of presentations, such as online video, context-sensitive help, viewgraphs, etc.		
Schedule. This topic area should include the training schedule.		

VERSION DESCRIPTION TEMPLATE		
Identification Number	**Title**	**Purpose**
NARA-IT-T-0014	Product/Version Description	The Product Version Description describes the components and interfaces of the system software and hardware configuration as it is built for each maintained version of the system.
Content Description		
The contents of a Product/Version Description should include the following topics:		
Introduction. The introduction sets the context of the document by identifying the Product Plan, Project Plan, and system to which it pertains, the background of the project, the scope of the development effort, an overview of the system, operational policies and constraints, document overview, relationship to other documents, and referenced documents.		
Executable Software. This topic should address the existence of application batch files, command files, data files, or other software files needed to install and operate the software on the target computers.		
Source Files. This topic should address the existence of source files for the application including any batch files, command files, data files, or other files needed to regenerate the executable software for the application.		
Compilation/build procedures. This topic should provide the procedures for compiling and building the executable files from source files and for preparation of the executable files for distribution.		
Startup / Initialization procedures. Include any special instructions for product initialization or implementation. Examples include adding a command to system startup files, components that require system administrator login, etc.		

USERS MANUAL TEMPLATE		
Identification Number	**Title**	**Purpose**
NARA-IT-T-0015	Users Manual	The Users Manual describes how to install and/or access and use a software and/or hardware system.
Content Description		
The contents of a Users Manual should include the following topics:		
Introduction. The introduction sets the context of the document by identifying the Product Plan, Project Plan, and system to which it pertains, the background of the project, the scope of the development effort, an overview of the system, operational policies and constraints, document overview, relationship to other documents, and referenced documents.		
System Users Manual. This topic area should include the following considerations: • Installation • Configuration parameter setup		
Software Users Manual This topic area should include the following considerations: • Software Application/Use • Software Organization and overview of operation • Access to the Software • Troubleshooting and Error Recovery • Assistance and problem reporting		

OPERATIONS MANUAL TEMPLATE		
Identification Number	**Title**	**Purpose**
NARA-IT-T-0016	Operations Manual	An Operations Manual provides information needed to operate a given computer, its software, and the peripheral equipment.
Content Description		
The contents of an Operations Manual should include the following topics:		
Introduction. The introduction sets the context of the document by identifying the Product Plan, Project Plan, and system to which it pertains, the background of the project, the scope of the development effort, an overview of the system, operational policies and constraints, document overview, relationship to other documents, and referenced documents.		
Hardware Operations. This topic area should include the following considerations: • Computer system preparation and shutdown • Operating Procedures • Diagnostic features • Diagnostic tools • Disaster Recovery Procedures		
Software Operations. This topic area should include the following considerations: • Application Use • Installation and Setup • Troubleshooting and Error Recovery • Software Input/Output Descriptions • Disaster Recovery Procedures		
Security. This topic area should include a description of the physical and automated access controls associated with the hardware, software, and communications components of the system.		
Facility Operations. Include any special instructions required for facilities operations such as power outage procedures, security access, and coordination of offsite pickups and deliveries.		

9 Deployment and Acceptance Testing

9.1 Purpose

Upon successful completion of the system test, the development team begins the deployment of the system into an operational setting. In some cases a partial deployment may occur to conduct the acceptance testing in a testbed type of setting (such as when networked applications need to be validated from multiple sites). The exact method of deployment should have been specified in the transition and installation plan.

Acceptance testing involves the users and operators in understanding system usage and ensures the system is delivered and performing as intended. If the system development process has been adhered to and users have been kept involved, there should be few changes to the needed. To some degree, the testing should simply be a formality and the activity becomes focused more on training and preparing the start of operations.

Significant discrepancies or system failures (e.g., they require rework of multiple system components as opposed to a single program) during Acceptance Testing may reflect that project success is in jeopardy. Unless the Product Owner is willing to accept corrections in a later release, the project will need to revert to an earlier activity to address them thus delaying the beginning of operations.

9.2 Deployment and Acceptance Testing Task Outline

The following timeline figure shows the typical sequence of Deployment and Acceptance Test tasks and work product creation.

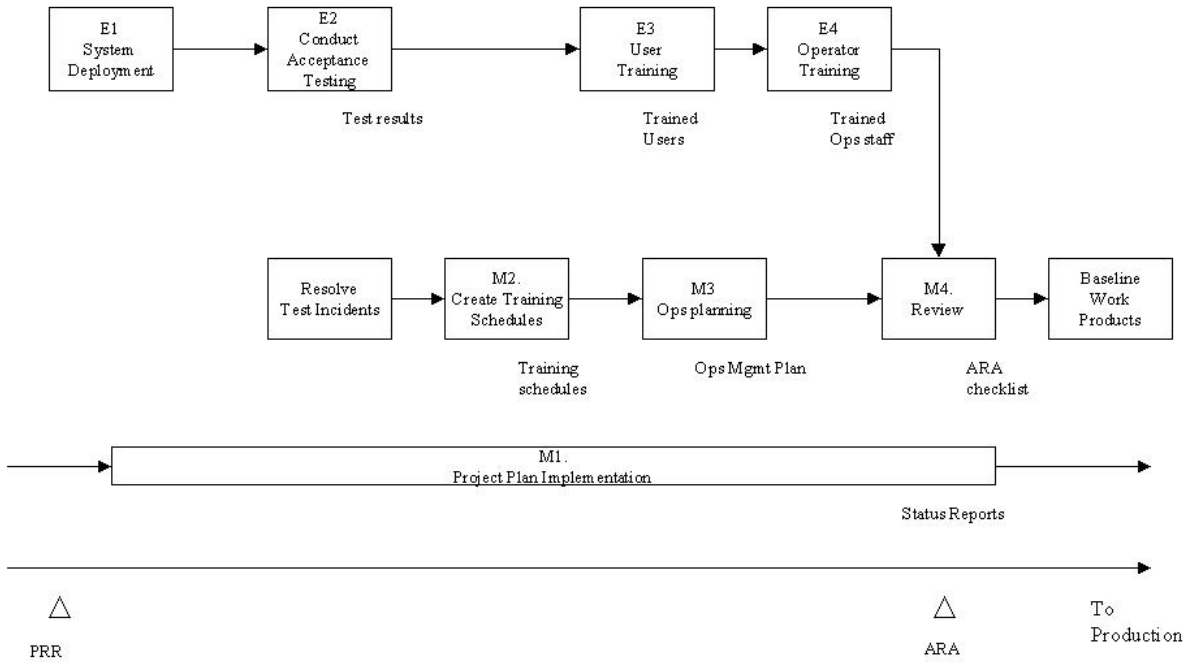

Figure 9-1 Deployment and Acceptance Testing Timeline

Table 9-1 Deployment and Acceptance Testing Performance Requirements

Entry Criteria	Deployment and Acceptance Testing Entry Criteria
Product Products and activities to be completed before beginning a new software development activity.	System Test completed successfully Acceptance Test Plan approved by Product Owner User and Operator materials prepared and approved by Product Owner and Operations System transition plan approved by Product Owner and Operations Resource allocation changes approved by Guidance Team Issues from Production Readiness Review resolved and documented Detail schedule prepared for Acceptance Testing and Transition tasks
Work Products	**Deployment and Acceptance Testing Work Products**
Products and activities to be completed during the current software development activity.	System Acceptance test report Ops Management Plan User and Operator training results Configuration Items System is scheduled or a draft schedule submitted to NWML for review and approval by the Archivist System meets general and system-specific records management requirements (as identified during Requirements Definition Phase)
Exit Criteria	**Deployment and Acceptance Testing Exit Criteria**
What must be finished before an ongoing activity is designated complete.	Acceptance Test completed successfully Operations Management Plan approved by Product Owner, Operations, and Guidance Team Users and Operators trained Detail Schedule approved for Production tasks Work product baselines established and delivered to Government CM manager

Engineering Tasks

E1. Deploy system in accordance with Installation and Transition Plan
   a. Prepare CI's for delivery and installation
   b. Establish environment for conducting acceptance test

E2. Conduct System Acceptance test in accordance with defined plan
   a. Prepare test data
   b. Run tests
   c. Log test results
   d. Resolve Test Incidents
   e. Integrate patches into system baselines
   f. Compare the collected test, inspection, or review results with established evaluation criteria to assess the degree of success.
   g. Assemble final package of CI deliverables and update deployed system

E3. Provide User Training
   a. Test users
   b. Assess results

E4. Provide Operator Training
   a. Test operators
   b. Assess results

Management Tasks

M1. Implement Project Plan elements for Deployment & Acceptance Testing
   a. Monitor schedule / Prepare detail for next activity
   b. Perform risk management
   c. Generate status reports

M2. Manage corrections / schedules for Test Incidents
   a. Negotiate acceptance of items that may be included in post production release
   b. Revise operational readiness justification

M3. Prepare User and Operator training schedules

M4. Operations Planning
   a. Initiate ops staffing
   b. Complete Ops Management Plan
   c. Define monitoring activity
   d. Status reporting
   e. Production Review schedules

M5. Perform Acceptance Review and Audit (ARA)
   a. Prepare and Distribute ARA materials (checklist, work products)
   b. Review materials prior to meeting
   c. Conduct ARA meeting
   d. Resolve open issues
   e. Establish configuration baselines for work products

Roles	Responsibilities
Product Owner	• Participates in Acceptance Testing (E2) • Identifies users for training / Attends Training (M2 / E2) • Signifies acceptance of the system (M4)
Guidance Team	• Approves system for production (M4)
Project Management	• Implements project plan elements (M1) • Updates operational readiness justification (M2) • Schedules user and operator training (M3) • Conducts ARA (M4) • Completes ARA checklist (M4)
Systems Engineering	• Manages system installation in accordance with plan (E1) • Prepares Systems and Facilities for System Acceptance testing (E1) • Participates in giving user and operator training (E3 / E4)
Data Management	• Prepares data environments for System Acceptance testing (E1) • Participates in giving user and operator training (E3 / E4)
Software Engineering	• Resolves software Test Incident reports (E1) • Participates in user and operator training (E3 / E4)
CM	• Prepares CIs for deployment • Integrates system fixes into build (E2) • Assembles final CI packages for delivery to Government CM manager • Documents change history (M1)
QA	• Ensures resolution of Test Incidents (E1) • Confirms users and operators trained and ready for production (E3 / E4) • Ensures completion of SA checklist (M4)
Test and Evaluation	• Conducts System Acceptance testing (E2) • Prepares Test Incident reports (E2) • Assists user / operator training (E3 / E4) • Identifies applicable elements to SA review checklist (M4)
Operations and Maintenance	• Participates in acceptance testing (E2) • Prepares Operations Management Plan (M3) • Attends Operator Training (E3)

9.3 Acceptance Review And Audit Checklist

Purpose and Objectives:

ARA is a final milestone prior to system production activity to ensure that acceptance criteria defined by the Product Owner has been achieved. In some cases the ARA is something of a formality and may be similar to the Production Readiness Review (PRR) performed earlier, but with an emphasis on readiness to start production.

Objectives for the ARA include

- Demonstrate that all acceptance criteria has been met
- Demonstrate that users and operators have been trained and are ready to begin use of the new system
- Confirm that external systems are ready to provide input or accept output

Significant discrepancies or system failures (e.g., they require rework of multiple system components as opposed to a single program) during Acceptance Testing may reflect that project success is in jeopardy. Unless the Product Owner is willing to accept corrections in a later release, the project may need to revert to an earlier development activity to address them.

Presenters

Project Management
Systems Engineering
Data Management
Software Engineering
Test and Evaluation

Participants

Product Owner(s)
Guidance Team
Operations Team
Additional user representatives when appropriate
QA

When:

After all tasks for Deployment and Acceptance are complete up to review. The review should take place as indicated in the Project Plan a specified number of days prior to operations to allow baselines of all deliverables to be set and the Installation and Transition plan to be implemented (30 - 90 days prior to operations is suggested).

Key Issues

- Have the Product Owners acceptance criteria been met?
- Are external systems ready to begin providing input and accepting output?
- Is the operations staff prepared to support the new release?

Entry Criteria

The following items should be done prior to conducting the review

- Distribution of Acceptance Test results, User Training results, and Operator Training results 1 week prior to review.

Exit Criteria

The review process is not considered closed until the following items are complete.

- All issues raised during the review have a documented resolution
- Transition to production approved by Product Owner and Guidance Team
- QA confirms completion of the Review Checklist

Presentation Agenda

Summary of changes / actions since Production Readiness Review
Acceptance Test results
User Training results
Operator Training results
External Interface Agreement status
Schedule for Production tasks

Checklist

- Do changes made to the system since the PRR create any risks that have not been addressed?
- Do Acceptance Test results validate that the system is ready to begin Production?
- Do user and operator training results indicate a sufficient understanding of the new system release to begin production?
- Are all external systems prepared for the start of operations?
- Is the schedule for Production tasks realistic?

9.4 Deployment and Acceptance Test Product Templates

ACCEPTANCE TEST REPORT TEMPLATE		
Identification Number	**Title**	**Purpose**
NARA-IT-T-0017	Acceptance Test Report	The Acceptance Test Report describes the NARA Project's test results and the recommendation for acceptance of the system for delivery from the developer.
Content Description		
The contents of an Acceptance Test Report should include the following topics:		
Introduction. The introduction sets the context of the document by identifying the Product Plan, Project Plan, and system to which it pertains, the background of the project, the scope of the development effort, an overview of the system, operational policies and constraints, document overview, relationship to other documents, and referenced documents.		
Test Summary Report. This topic area should summarize the results of the designated testing activities and provide evaluations based upon the results. Analysis of the results should be done according to the project's Test Strategy.		
Test Incident Report. This topic area should document any event that occurs during testing which requires investigation.		
Test Log. This topic area should include a chronological record of relevant details about the execution of tests.		
Acceptance Recommendation. This topic area should include the final recommendation for acceptance of the system including acceptance with specific changes/fixes.		

10 Production

10.1 Purpose

The Production tasks ensure that operations are initiated and then take place as planned. This may require clearing of any remnants of acceptance testing.

Throughout the system's life there will inevitably be problems that require fixes, enhancements desired by users, and changes driven by the release of new software or other system components some of which may no longer be supported by a vendor.

Management monitors operational performance to determine whether problems exist, to coordinate changes and upgrades with users, and generate operational status reports per NARA or other government directives. Reviews are conducted on a routine basis as defined in the Ops Management Plan.

10.2 Production Task Outline

The following timeline figure shows the typical sequence of Production tasks and work product creation.

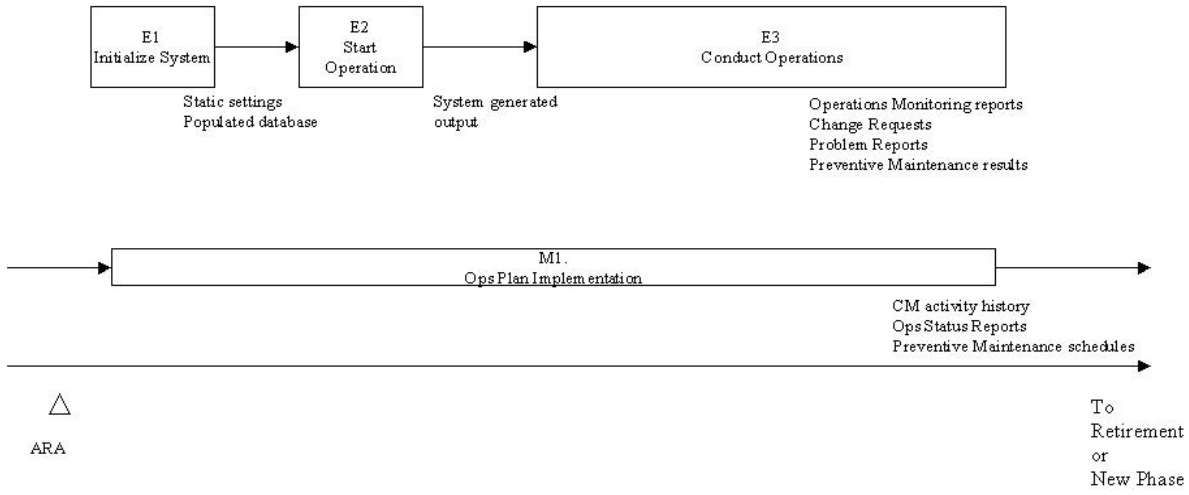

Figure 10-1 Production Timeline

Table 10-1 Production Performance Requirements

Entry Criteria	Production Entry Criteria
Products and activities to be completed before beginning a new software development activity.	Acceptance Testing completed successfully Operations Management Plan approved by Product Owner, Operations, and Guidance Team Users and Operators trained Detail Schedule for Production initiation tasks
Work Products	**Production Work Products**
Products and activities to be completed during the current software development activity.	Continuous Monitoring / Problem Reports / Resolutions Change Requests System Revisions (New phase definitions) Status Reports System enhancements are reviewed for EA and standards compliance Performance metrics are tracked, monitored and reviewed against Capital Planning expectations. Configuration Management and Control Records schedule has been validated or modified to reflect the findings of the assessment. Records management requirements have been validated or modified and then recertified by the NARA Records Officer.
Exit Criteria	**Production Exit Criteria**
What must be finished before an ongoing activity is designated complete.	Retirement or Rollover

Engineering Tasks

E1. Initialize system in accordance with Installation and Transition plan
   a. Install new components
   b. Migrate / populate database per plan
   c. Set initialization values as required

E2. Start System Operation
   a. Start system components per Operations Manual
   b. Review production output to ensure system is operating as expected

E3. Perform Operations and Maintenance
   a. Generate operational materials in accordance with Ops Management Plan
   b. Respond to Problem Reports
      i. Identify source of problem
      ii. Take appropriate actions to correct the problem
      iii. Create patch release for Configuration Item(s) to correct system error
   c. Respond to Change Requests
      i. Provide impact analysis
      ii. Review with CCB for development approval / Identify target system release
      iii. Enter development phase
   d. Perform preventive maintenance
      i. Identify maintenance requirements
      ii. Assess expected outage periods
      iii. Notify users of PM activities

Management Tasks

M1. Operations Management
   a. Monitor Operations and prepare status reports in accordance with the Ops Management Plan
   b. Facilitate CCB meetings
   c. Establish change requests to be included in new phase / system release
   d. Schedule and coordinate preventive maintenance and other planned outages with users
   e. Conduct Production Performance Reviews per Ops Management Plan

Roles	Responsibilities
Product Owner	• Submits / evaluates Change Requests • Participates in CCB • Ensures system operations are conducted in accordance with Operations Plan • Prepares status reports • Facilitates CCB • Prepares resource impact assessments of Change Requests • Allocates Change Requests to new system releases / phases • Prepares System Retirement or Rollover Plans
Guidance Team	• Approves resource allocation of Change Requests to system releases / development phases
Systems Engineering	• Initializes production system • Resolves facilities and hardware Problem Reports • Prepares technical impact assessments of Change Requests
Data Management	• Initializes (populates) production databases • Monitors performance of database systems • Resolves data management Problem Reports • Performs impact analysis for Change Requests
Software Engineering	• Resolves software Problem Reports • Performs impact analysis for Change Requests
CM	• Integrates system fixes into build (i.e., prepares system patches) • Documents change history
QA	• Ensures resolution of Problem Reports in accordance with Operations Plan
Test and Evaluation	• Performs systems testing of build patches
Operations and Maintenance	• Ensures processing occurs in accordance with Operations Plan • Provides User Help Desk support • Prepares Problem Reports • Generates O&M Reports

10.3 Production Performance Review Checklist

Purpose and Objectives:

Production Performance Reviews are conducted on a recurring basis to assess whether the system continues to meet the specified System Requirements. This is especially true for systems that grow during their use through the accumulation of data or the addition of users.

Presenters

Ops Management
Systems Engineering / Administration

Participants

Product Owner(s)
Guidance Team

When:

Defined in the Operations Management Plan. It is likely to occur on monthly, quarterly, or semi-annual basis depending on the scale of the system and growth in usage.

Key Issues

- Does the system continue to meet requirements?
- Is the system being used as anticipated? If not, what needs to occur to meet system goals?

Entry Criteria

The following items should be done prior to conducting the review

- Operations status reports are distributed
- Summary of problem reports and change requests

Exit Criteria

The review process is not considered closed until the following items are complete.

- Action plans for needed operations adjustments have been developed
- Issues raised during reviews have a documented resolution

Presentation of Production Status

This section should be completed based on the needs of the Production system. Include the following:

Status of Change Requests
Status of Problem Reports

Checklist

This section should be completed based on the needs of the Production system.

11 Retirement and Rollover

11.1 Purpose

Retirement and Rollover tasks ensure that important elements of the system are retained for future use. This may include migration of data to a new system or merely retaining products for institutional knowledge.

In any case certain steps should be followed to ensure all final inputs are processed through the system and that products are inventoried and retained as required.

11.2 Task Outline

The following timeline figure shows the typical sequence of Requirements Definition tasks and work product creation.

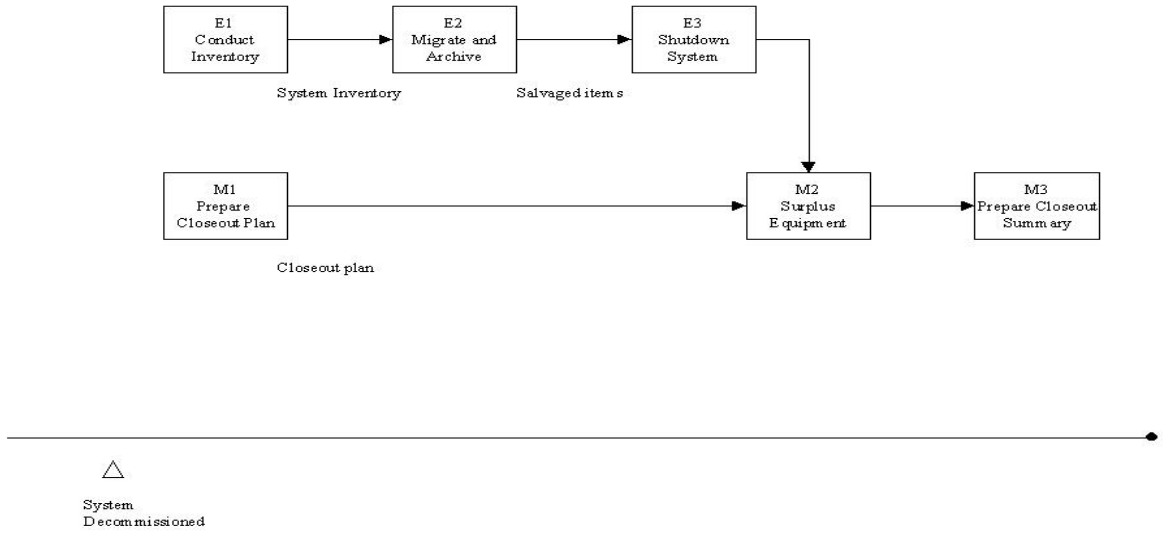

Figure 11-1 Retirement / Rollover Timeline

Table 11-1 Retirement/Rollover Performance Requirements

Entry Criteria	Retirement/Rollover Entry Criteria
Products and activities to be completed before beginning a new software development activity.	Replacement system ready for production -OR- System mission complete Retirement / Rollover Plan accepted
Work Products	**Retirement/Rollover Work Products**
Products and activities to be completed during the current software development activity.	System retirement plan including media sanitization System Closeout / Lessons Learned Report Saved or migrated system products (i.e., information preservation) System retirement is identified in the EA Sequencing Plan All system integration ramifications are identified and have a remediation strategy.
Exit Criteria	**Retirement/Rollover Exit Criteria**
What must be finished before an ongoing activity is designated complete	System retirement plan complete including Hardware / Software disposal, data migration, retention or authorized destruction.

Engineering Tasks

E1. Identify all artifacts of the system that should be migrated, archived, or otherwise maintained to include
   a. data,
   b. documents,
   c. software
   d. other valid system components
E2. Perform migration activities
E3. Shutdown System

Management Tasks

M1. Prepare and implement system retirement plan
M2. Perform surplus equipment tasks
M3. Prepare System Closeout / Lessons Learned summary

Roles	Responsibilities
Product Owner	Sets system retirement agenda
Guidance Team	Approves System Retirement plan Identifies system components to be retained
Operations Management	Implements System Retirement plan Prepares Closeout report
Systems Engineering	Prepares system components for surplus or archival
Data Management	Prepares data for migration or archival
Software Engineering	Prepares software components for migration or archival
CM	Prepares work products for migration or archival
QA	Ensures conformance to and completion of Retirement Plan
Test and Evaluation	Reviews retirement plan
Operations and Maintenance	Supports migration and archival activities

12 General Product Templates (Not specific to Activity)

WHITE PAPER TEMPLATE		
Identification Number	**Title**	**Purpose**
NARA-IT-T-0003	White Paper	A White Paper is an informal document that presents an idea for consideration. It may be a pre-cursor to a Concept of Operations Document, a Product Plan, or any other document, and may be written in any phase of a development lifecycle. White papers are most often written before development or finalization of a Concept of Operations.
Content Description		
The contents of a White Paper should include the following topics:		
Introduction. The introduction sets the context of the paper by identifying the purpose, scope, and background of the issues and recommendations.		
Description. This section should describe the main topic of the paper. Topics may include, but are not limited to, an analysis of alternative approaches for system solutions, a system boundary or scoping suggestion, recommended procedural, data sharing, or interoperability improvements, recommendations for improvement to a specific automation problem, or a problem that could be resolved using an automated solution.		
Conclusions/Recommendations. This section should include a summary of the conclusions and/or recommendations concerning the topic.		

CAPABILITY EVALUATION REPORT TEMPLATE		
Identification Number	**Title**	**Purpose**
NARA-IT-T-0018	Capability Evaluation Report	The Capability Evaluation Report provides a technical and business perspective for meeting the objectives of a specific MED phase and provides a go/no-go recommendation to the next planned phase.
Content Description		
The contents of a Capability Evaluation Report should include the following topics:		
Introduction. The introduction sets the context of the document by identifying the Product Plan, Project Plan, and system to which it pertains, the background of the project, the scope of the development effort, an overview of the system, operational policies and constraints, document overview, relationship to other documents, and referenced documents.		
Capability Completion Objectives. This topic includes a list of objectives for a specific MED phase product which might be a testbed, prototype, pilot, or APU system.		
Capability Results. This topic addresses the evaluation of the end product against its objectives for a specific MED phase.		
Lessons Learned. This topic addresses detailed lessons learned from producing and evaluating the MED phase product.		
Next-Step Recommendation. This topic summarizes the recommendation for moving to the next planned MED phase, or re-planning a MED phase based upon lessons learned.		
Approvals. This topic provides formal sign-off of Project approval authorities for moving to the next MED phase.		

13 Selected Readings

Davis, Alan M., Software Requirements - Objects, Functions, and States

Jones, T. Capers, Estimating Software Costs

Rechtin, E. and Maier, M., The Art of Systems Architecting

Perry, William E., Effective Methods for Software Testing

Grady, Jeffrey O., System Requirements Analysis

Royce, Walker, Software Project Management - A Unified Framework

Information Technology Resources Board, Assessing the Risks of Commercial-Off-the-Shelf Applications (online)

NASA Software Engineering Laboratory, Manager's Handbook for Software Development (online 84-101.pdf)

NASA Software Engineering Laboratory, SEL Package-Based System Development Process (online)

NASA SP-610S, Systems Engineering Handbook (online)

Carnegie-Mellon University Software Engineering Institute, A Systems Engineering Capability Maturity Model (online)

CMU - SEI, Capability Maturity Model - Integrated Systems / Software Engineering (online)

Federal ROI and Cost-Benefit Analysis Guidelines http://www.cio.gov/files/roi.pdf and http://www.whitehouse.gov/OMB/circulars/a094/a094.html