CSR
NRFC0004 / BCP0003
Category: Best Community Practice

R. McKay
CSR
January 2006

National Institutes of Health

Systems Development Life Cycle Framework

v1.0

Status of this Memo

This document specifies a best community practice (BCP) for the National Institutes of Health (NIH) and requests discussion and suggestions for improvements. Distribution of this memo is unlimited.

Abstract

The purpose of this document is to specify the role of the NIH Enterprise Architecture in the systems development life cycle for enterprise systems at the National Institutes of Health (NIH). It also provides background information and an overview of systems development life cycle (SDLC) concepts. The concepts presented may serve as a foundation for the life cycle management approach adopted by organizations managing enterprise systems at or for NIH.

Table of Contents

1. Background and Enterprise Architecture Process Requirements

Without a methodology systems development becomes haphazard and subsequently a risky and expensive undertaking in terms of cost, schedule, and quality. To mitigate this risk NIH established the following enterprise architecture principle:

> "Developers and maintainers of enterprise applications will have a documented systems development life cycle (SDLC)."

Though it is not in the scope of the NIH Enterprise Architecture to mandate an agency-wide SDLC, this document describes a traditional SDLC that organizations may use as a starting point for developing and documenting SDLCs for enterprise systems. Organizations responsible for implementing and maintaining enterprise systems are responsible for the documentation, publication, and management of their local SDLC.

SDLCs for all enterprise systems must include the following enterprise architecture processes steps in the SDLCs for new system implementation or release management (see Figure 1):

- **System Concept Development Phase**: contribute to the system definition
- **Planning Phase**: provide patterns and bricks
- **Requirements Analysis Phase**: provide enterprise architecture (EA) requirements and review final document
- **Design Phase**: review integration and technology, the data model, and interface definition
- **Development Phase**: none
- **Integration and Test Phase**: participate in the final integration review
- **Implementation Phase**: participate in the post implementation review
- **Operations and Maintenance Phase**: check system during each enhancement cycle
- **Disposition Phase**: none

For those that use a more iterative style of SDLC, please reference Figure 2 to view the impact of EA on this type of SDLC.

In both diagrams, note the placement of outside drivers and their interaction with the SDLC:

- CPIC – Capital Planning and Investment Control. At designated milestones in the life cycle of a project/system, analysis is conducted on the investment feasibility of the IT project/system.
- C&A – Certification and Accreditation. Before implementation of an IT system, an audit of the system security controls is conducted to certify that they are in accordance with mandated guidelines.
- FSMA – Federal Security Management Act. The status of security of all implemented IT systems must be reported on an annual basis to OMB.

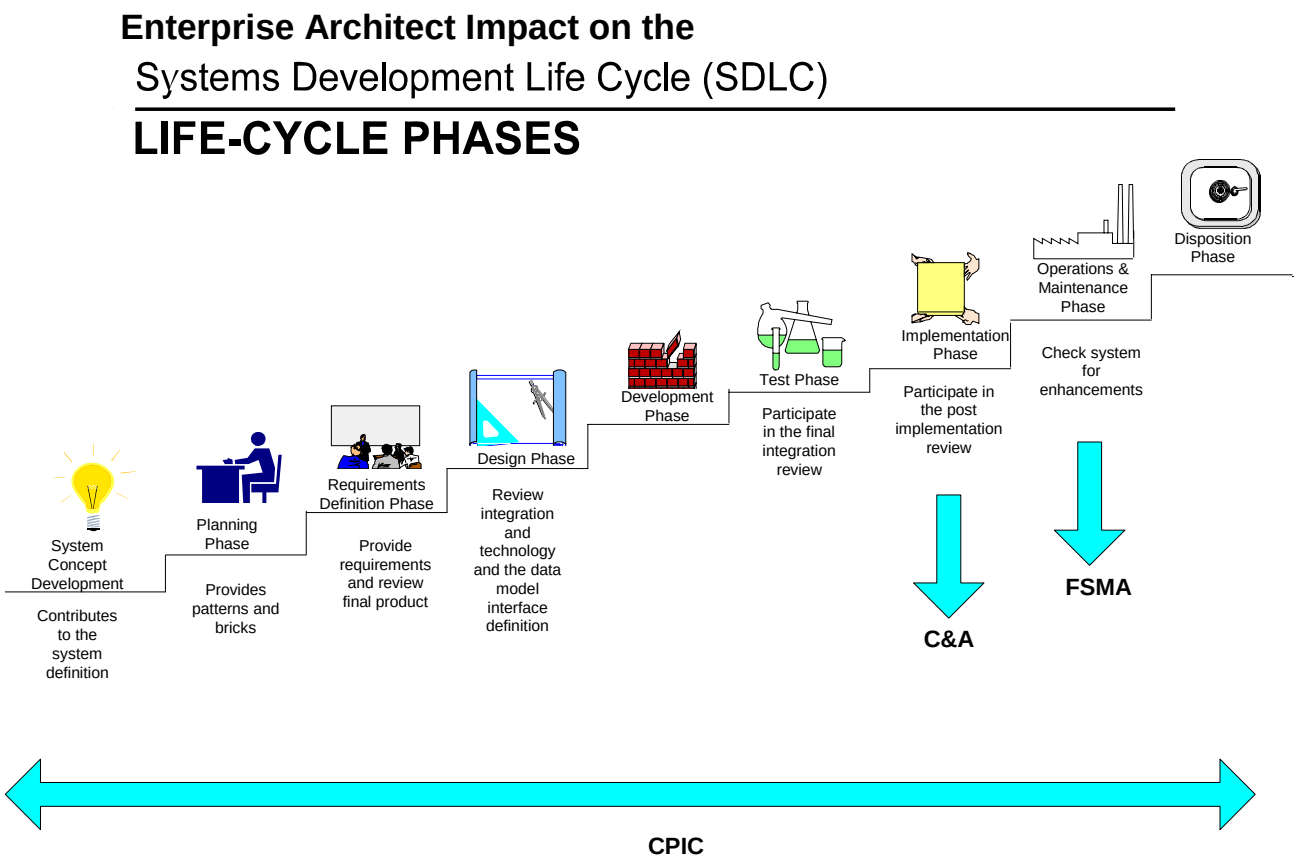


Figure 1: Enterprise Architecture Impact on the Waterfall Type of Systems Development Life Cycle

Project teams must coordinate with the Office of the Chief IT Architect (OCITA) at the beginning of each phase as described at enterprisearchitecture.nih.gov.

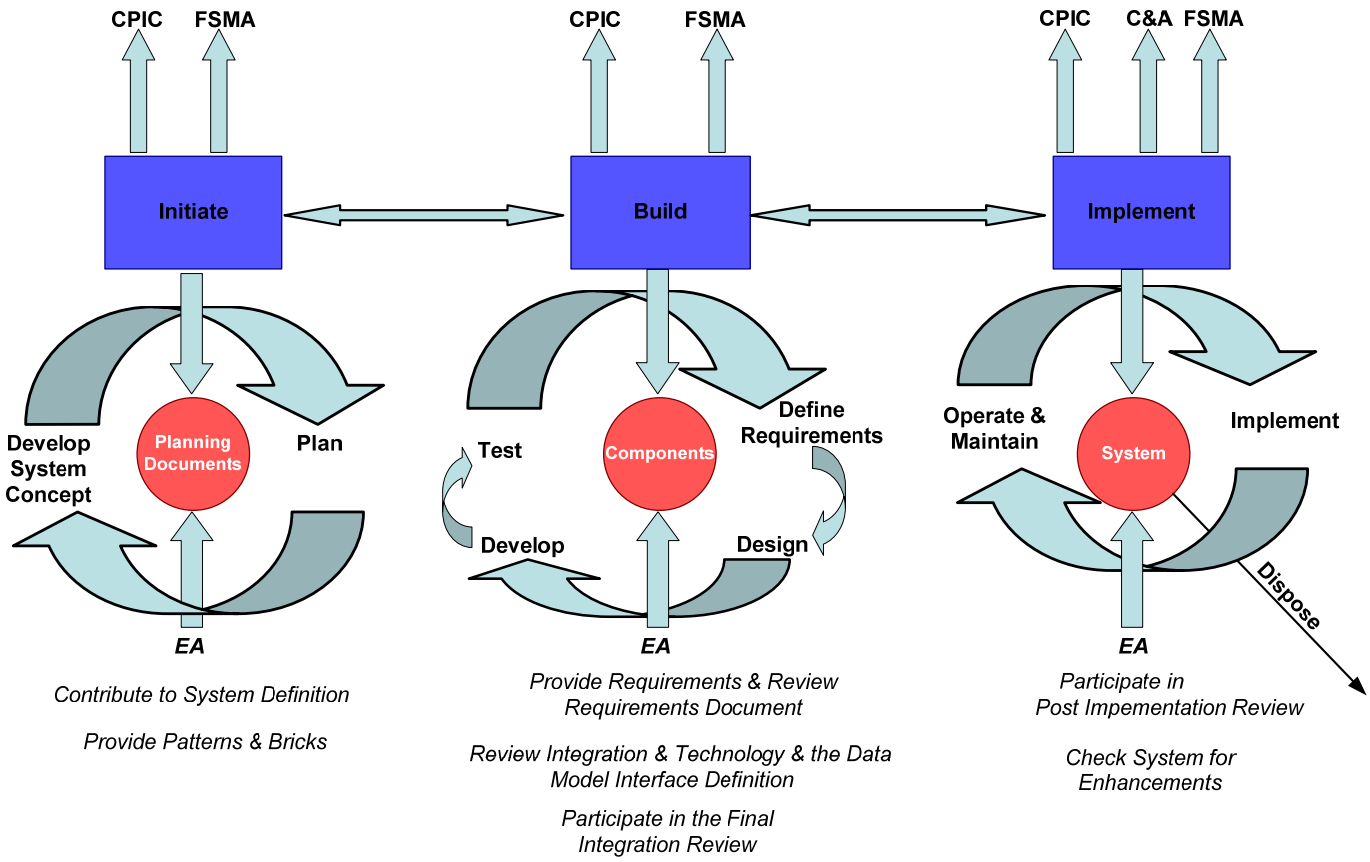


Figure 2: Enterprise Architecture Impact on the Iterative Type of Systems Development Life Cycle

2. Introduction to the Systems Development Life Cycle: The Nine Phases of the SDLC Framework

The SDLC framework describes a broad and diverse set of activities for addressing information systems efforts. This framework will help address the needs of other organizations tasked with the following associated responsibilities:

- Ensuring available funding for development and system life-cycle costs
- Ensuring the security and integrity of the system
- Maintaining the system until system retirement
- Training system users
- Maintaining and disposing of NIH records

The full life-cycle model is divided into nine phases. Depending on the size and complexity of a project, alternative work patterns may be selected that will result in the combining or overlapping of specific phases. Not every project will require that every phase be executed. In Exhibit 1, Life Cycle phases, all nine phases are presented depicting the traditional, full, sequential work pattern.

Note: In the iterative type of SDLC, a set of phases may be completed for each component, as the system is built up consecutively of components.

2.1 System Concept Development Phase

System concept development actually starts the life cycle when a need to develop or significantly change a system is identified. Once a business need, based on operational requirements, is identified and documented the approaches for meeting it must be reviewed for feasibility and appropriateness. The need may involve development of a new system or modification of an existing system. Approvals and funding are needed before beginning the Planning phase.

2.2 Planning Phase

The Planning phase begins after the project has been defined and resources have been committed to the project. A project plan is developed that documents the approach to be used and includes the discussion of methods, tools, tasks, resources, project schedules, and user input.

2.3 Requirements Definition Phase

Functional user requirements are formally defined in a Functional Requirements Document (FRD). The requirements are delineated in terms of data, system performance, security, and maintainability requirements for the system. All requirements are defined to a level of detail sufficient for systems design to proceed. See Exhibit 2 for a template for the FRD.

2.4 Design Phase

The external physical characteristics of the system are designed during this phase. The operating environment is established, major subsystems and their inputs and outputs are defined, and

processes are allocated to resources. Everything requiring user input or approval must be documented and reviewed by the user. The internal physical characteristics of the system are specified and a detailed design is prepared. Subsystems defined during the external design are used to create a detailed structure of the system. Each subsystem is partitioned into one or more design units or modules. Detailed logic specifications are prepared for each module. The Design phase ends with a formal design walk-through with the user and approval of the design by the System Owner.

Exhibit 1:

Systems Development Life Cycle (SDLC)

LIFE-CYCLE PHASES

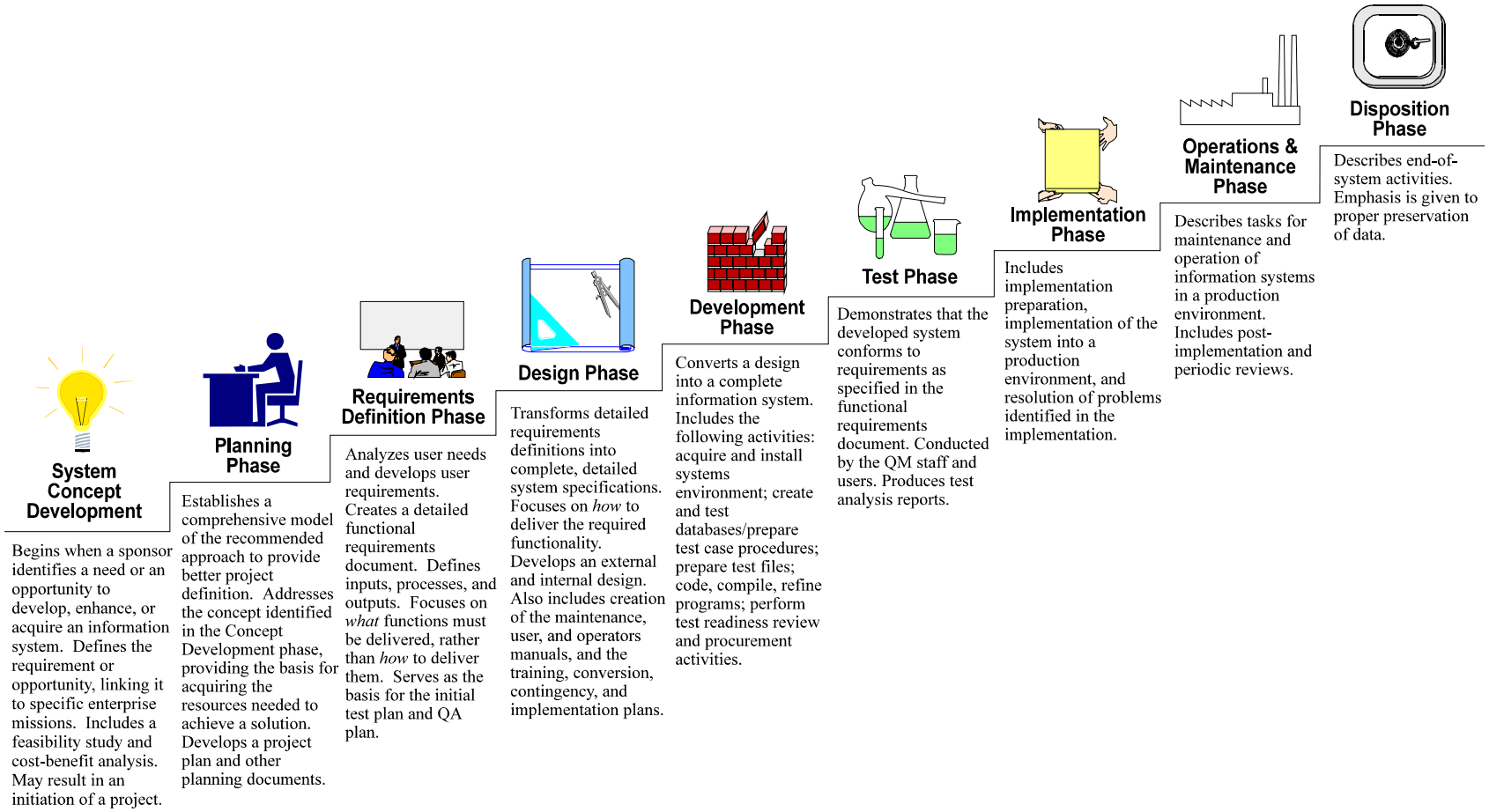


Exhibit 2: Functional Requirements Document Outline

Cover Page

Table of Contents

1. Introduction
- 1.1 Project Description
  - 1.1.1 Background
  - 1.1.2 Purpose
  - 1.1.3 Assumptions and Constraints
  - 1.1.4 Interfaces to External Systems
- 1.2 Points of Contact
- 1.3 Document References

2. Business Requirements
- 2.1 Data Requirements
- 2.2 Functional Process Requirements

3. Operational Requirements
- 3.1 Security
- 3.2 Audit Trail
- 3.3 Data Currency
- 3.4 Reliability
- 3.5 Recoverability
- 3.6 System Availability
- 3.7 Fault Tolerance
- 3.8 Performance
- 3.9 Capacity
- 3.10 Data Retention

4. Requirements Traceability Matrix

5. Concepts of Operations (CONOPS)
- 5.1 Definition of Features
- 5.2 Description of Operations
- 5.3 User Organization View
- 5.4 Effect on Operations and Personnel
- 5.5 Effect on Existing Operations
- 5.6 Interfaces to Other Systems
- 5.7 Methods of Implementation
- 5.8 Figure (Optional)

APPENDIX A—GLOSSARY

2.5 Development Phase

The detailed specifications produced during internal design are translated to executable software. Software is unit tested, integrated, and retested in a systematic manner. Hardware is assembled and tested.

2.6 Test Phase

Subsystem integration, system, security, and user acceptance testing are all conducted during the Test phase. The user, with the Quality Assurance (QA) organization, validates that the functional requirements, as defined in the functional requirements document (FRD), are satisfied by the developed or modified system.

2.7 Implementation Phase

The system or system modifications are installed and made operational in a production environment. The phase is initiated after the system has been tested and accepted by the user. The phase continues until the system is operating in production in accordance with the defined user requirements.

2.8 Operations and Maintenance Phase

The system operation is ongoing. The system is monitored for continued performance in accordance with user requirements, and needed system modifications are incorporated. Operations continue as long as the system can be effectively adapted to respond to an organization's needs. When modifications are necessary, the system may reenter the planning phase, depending on the size and nature of the modification.

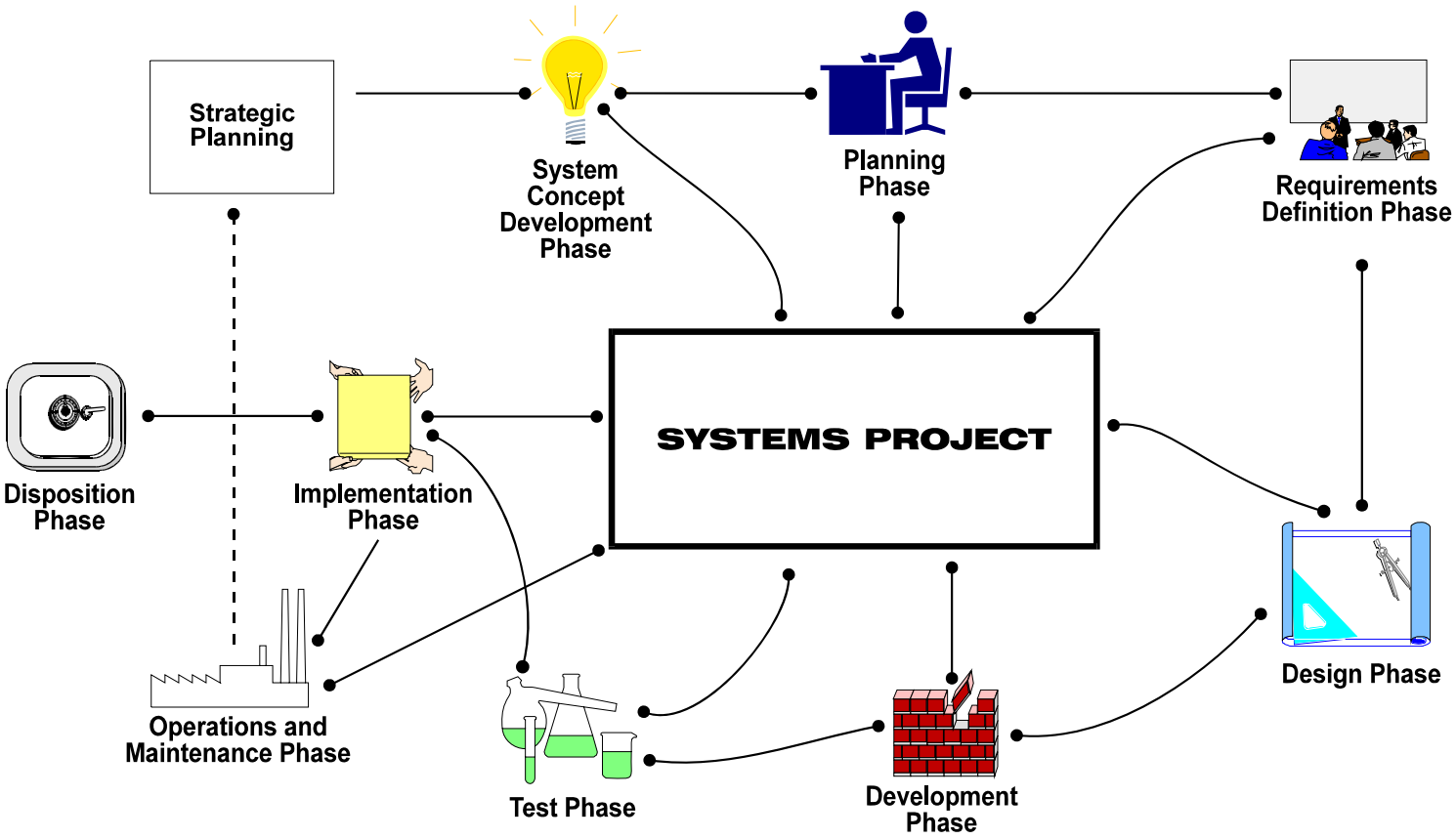
2.9 Disposition Phase

The disposition activities ensure the orderly termination of the system and preserve the vital information about the system so that some or all the information may be reactivated in the future if necessary. Particular emphasis is given to proper preservation of the data processed by the system so that the data are effectively migrated to another system or archived in accordance with applicable records management regulations and policies for potential future access.

2.10 Phase Interaction

Exhibit 3, Phase Interaction, shows the relationship among the life-cycle phases. Decisions reached and deliverables completed in one phase may be required in subsequent phases. As the project progresses, life-cycle documentation becomes an integral part of the development process. It may be necessary to cycle back to an earlier phase of the life cycle, particularly if an iterative design/development strategy is appropriate for the particular project. This strategy would be selected at the time of project initiation. Another example of looping back is to clear up problems that surface during testing – causing the project to move back into "development" again.

Exhibit 3: Phase Interaction



3. Documentation

The life-cycle methodology specifies which documentation will be generated during each phase. Some of the products may be the basis for the NIH information collection requirements, Information Resources Management (IRM) reviews, and cost-benefit analyses (CBAs).

The outputs of SDLC documentation activities are typically categorized into two major types: process and product, as follows:

- **Process Documentation**—Process documentation communicates status and direction. It addresses the actions required for developing, implementing, and maintaining the system. Process documentation is not updated after implementation; however, it should be retained for evaluation and general reference. Examples include project plans, timelines, funds required, procedures to be followed, project review reports, requirements documents, and design documents.
- **Product Documentation**—Product documentation describes the system itself, what it is, how it is operated, and how it is to be maintained. It is most often used by individuals who were not directly involved in the system's development and instructs them on how to effectively operate, maintain, and use the system. Modifications to the system are reflected in the documents as they occur and new versions are distributed periodically. Examples include user manuals, operations manuals, and maintenance manuals.

Some documentation remains unchanged throughout the systems life cycle while others evolve continuously during the life cycle. Other documents are revised to reflect the results of analyses performed in later phases. Each of the documents produced are collected and stored in a project file. It is the intent of this SDLC framework to provide a degree of flexibility in the documentation required for each new system. Very large system development projects will likely use a modular approach to development, delivering each module through a release strategy as it is completed. Small projects may utilize Joint Application Development, Rapid Application Development (RAD) and/or prototyping development methods. Either modular development or RAD would require a customized set of documentation. The specific list of documents and any modifications to the standard for each should be negotiated and agreed upon at the time of project initiation. Each of the documents should be reflected in the project plan. A list of deliverables by phase is provided in Exhibit 4, Phases and Deliverables.

Exhibit 4: Phases and Deliverables

Phases and Deliverables	System Concept Development	Planning	Requirements Definition	Design	Development	Test	Implementation	Operations and Maintenance	Disposition
Concept of Operations	C/F								
Cost-Benefit Analysis	C	R	R	R	R	R	R	F	
Feasibility Study	C/F								
Risk Management Plan	C	R	R	R	R	F			
Request for Information Services (RITS)	C/F								
Acquisition Plan		C	R	R	F				
Configuration Mangagement Plan		C	R	R	R	R	R	R	F
Privacy Act Federal Register Notice	C	F							
Project Plan		C	R	R	R	R	R	R	F
Records Disposition Schedule		C	R						
Change Control Documents			C	C	C	C	C	F	
Functional Requirements Document			C	F					
Quality Assurance Plan			C	R	F				
Security Risk Assessment			C	R	R	R	F		
Sensitive System Security Plan			C		R	R	R	F	
Test Plan			C	R	F				
Contingency Plan				C	R	R	R	R	
Conversion Plan				C	R	F			
External Design Document				C	F				
Implementation Plan				C	R	F	F		
Internal Design Document				C	F				

C=Created in this phase, R=Revised in this phase, F=Finalized in this phase

Exhibit 4: Phases and Deliverables (continued)

Phases and Deliverables (continued)	System Concept Development	Planning	Requirements Definition	Design	Development	Test	Implementation	Operations and Maintenance	Disposition
Maintenance Manual				C	R	F		R/F	
Operations Manual				C	R	F			
Training Plan				C	R	F			
User Manual				C	R	F			
Software Development Folder					C	R	F		
System Fielding Authorization					C		F		
System Software					C	R	F		
Test Files/Data					C	F	C/F		
Test Analysis Approval Determination						C/F			
Test Analysis Report						C/F			
Test Problem Reports						C/F			
Delivered System							C/F		
Security Accreditation Statement							C/F		
Security Certification Statement							C/F		
Periodic Review Report								C/F	
Postimplementation Review Report								C/F	
User Satisfaction Review								C/F	
Disposition Plan									C/F
Posttermination Review Report									C/F

C=Created in this phase, R=Revised in this phase, F=Finalized in this phase

4. Crosscutting Activities

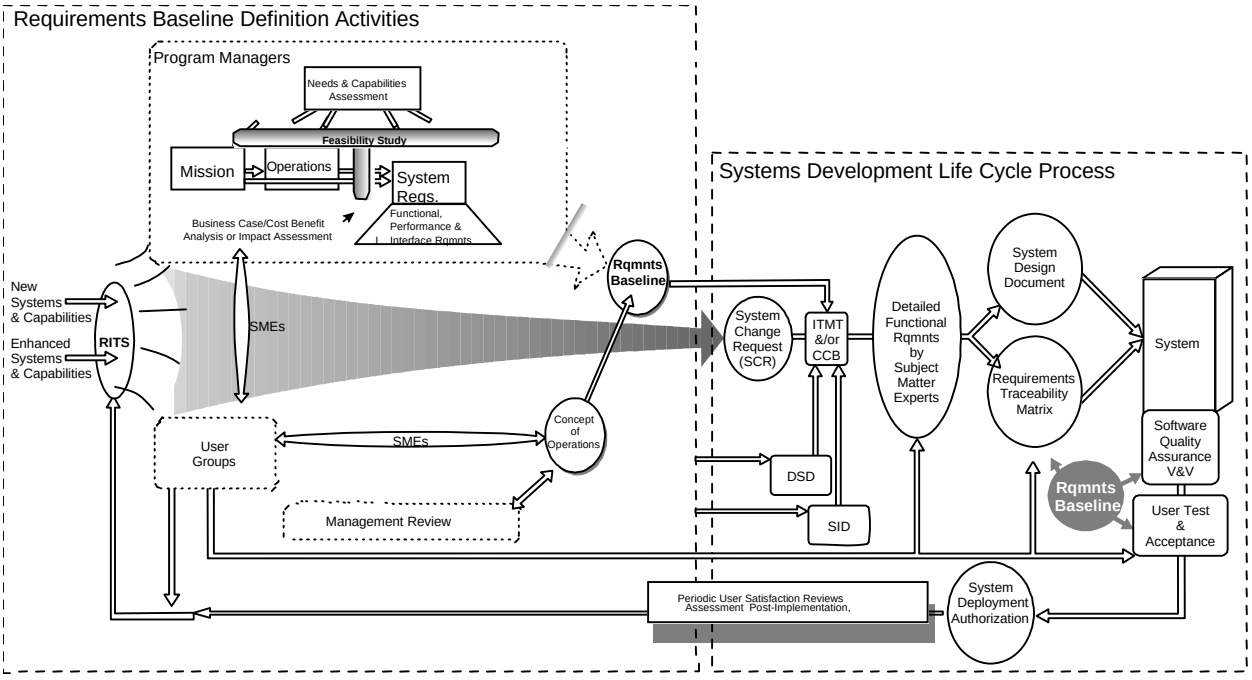
There are a number of activities that crosscut the phases of the systems life cycle. These activities are addressed and summarized in the subsequent sections.

4.1 Requirements Management

The purpose of requirements management is to establish and control the scope of system development efforts and facilitate a common understanding of system capabilities between the System Proponent, developers, and future users. The goal of the SDLC is that requirements for developed software be consistently documented and controlled and that software products conform to the requirements established.

The requirements management process, involves identifying, tracking, and validating system requirements. This process is shown in Exhibit 5, Overview of Requirements Management Process. Requirements management establishes and maintains a common understanding between a customer, NIH or NIH management, and the software development team and tightly controls the establishment, prioritization, acceptance of, and changes to documented requirements. This kind of control ensures that customers, management, and the development team are aware of, and can agree to, any changes before they are made and enables the development team to adjust plans and designs according to accepted changes.

Exhibit 5: Overview of Requirements Management Process



The SDLC framework requires the use of a requirements traceability matrix. An example of a matrix is shown in Exhibit 6, Sample Requirements Traceability Matrix.

Exhibit 6: Sample Requirements Traceability Matrix

Functional Requirement		*Verification Method				Test Plan Reference
Description	Para. Reference	A	I	D	T	
The functionality of the Enhanced Primary Verification Process **will be** an expansion of the functionality of the point of sale (POS) emulation logic that is currently in place to support primary verification queries to ASVI.	**3.2-01**				**X**	**TC 2.3.1.6**
The 200 employers who will be part of the Phase II TVS Pilot **will submit** data electronically via an interface with the ASVI system.	**3.2-02**				**X**	**TC 2.3.1.6**
All secondary information **will be** passed electronically to the LA FCO from ASVI for secondary verification.	**3.2-03**				**X**	**TC 2.3.1.10**
After a determination has been made on a case, the status verifier **will then** send the response back to the employer electronically; the return path is the exact opposite of the preceding path to the FCO.	**3.2-04**				**X**	**TC 2.3.1.10**
The new system **will be** capable of tracking information on each case throughout both the primary and secondary verification processes.	**3.2-05**				**X**	**TC 2.3.1.6, 2.3.1.7, 2.3.1.11**

A = ANALYSIS **I** = INSPECTION **D** = DEMONSTRATION **T** = TEST

4.2 Project Planning

Project planning is documented for every effort regardless of size, scope, or complexity. The project plan documents three important project aspects: the work approach, commitments, and estimates of the effort, as follows:

- **Work approach**—Includes documenting the project development strategy, selected alternative work pattern, adjustments to the life-cycle structure, selection of tools and methodologies, identification of applicable reviews and approvals, and project standards and procedures
- **Work estimates**—Includes documenting the schedule, staffing and resources needed, and an estimate of the size of the software work products to be developed
- **Commitments**—Including goals, constraints, and sponsorship

The project plan is first produced in the Planning phase and is updated, expanded, and refined continually throughout the life cycle. An example of a project plan is shown in Exhibit 7, Project Plan. The project plan may be subject to approval by management for large projects or projects under their oversight.

Exhibit 7: Project Plan Outline

PROJECT PLAN

Cover Page

Table of Contents

1. Introduction
- 1.1 Project Description
- 1.2 Project Background
  - 1.2.1 Project Development Strategy
  - 1.2.2 Organization of the Project Plan
- 1.3 Points of Contact
- 1.4 Project References
- 1.5 Glossary

2. Organization and Responsibilities

3. Project Description, Schedule, and Resources
- 3.1 Project Work Breakdown Structure
  - 3.1.1 Summary Work Breakdown Structure
  - 3.1.2 Project Work Breakdown Structure
  - 3.1.3 Contract Work Breakdown Structure
  - 3.1.4 Work Breakdown Structure Dictionary
- 3.2 Resource Estimates
- 3.3 Schedule
- 3.4 Resource Acquisition Plan
- 3.5 Communication Plan
- 3.6 Project Standards and Procedures
- 3.7 Risk Management

4.0 Security and Privacy
- 4.1 Privacy Issues
- 4.2 Computer Security Activities

4.3 Project Tracking and Oversight

The goal of project tracking and oversight are to provide visibility into the software development efforts, as described in the project plan. Project tracking and oversight includes reviewing accomplishments to date, comparing estimated to actual efforts, adjusting plans, and communicating the results. The project tracking and oversight process permits the system proponent to continually assess and manage risks, future resource requirements, and recommend corrective actions.

4.4 Configuration Management

Continual, consistent documentation of the development and evolution of the system is essential to ensure that at all points in the systems life cycle key analyses and decisions are recorded, and that the system is accurately described. The mechanisms for accomplishing this are performed by Configuration Management (CM). CM is a discipline consisting of the following four major elements:

- **Identification**—The selection and labeling of all functional and physical characteristics of the hardware and software
- **Control**—The evaluation, coordination, approval, and implementation of all approved changes to the contents of an established configuration baseline, that is, described characteristics of the hardware and software
- **Status accounting**—The administrative support required and communication of the status of the system throughout its life cycle
- **Configuration audits**—A formal examination of the configuration records and system documentation to verify that the system is accurately documented and that approved changes have been incorporated

CM provides a set of tools and procedures to control changes to a system after life-cycle activities have begun. The template for each project to use to create its own plan for accomplishing CM can be found in Exhibit 8, Configuration Management Plan.

Exhibit 8: Configuration Management Plan Outline

CONFIGURATION MANAGEMENT PLAN

Cover Page

Table of Contents

1. Introduction
- 1.1 Purpose
- 1.2 Scope
- 1.3 Policy
- 1.4 System Description
- 1.5 Definitions
- 1.6 Reference Documents

2. Organization
- 2.1 CM Activities
- 2.2 CM Responsibilities

3. Configuration Identification
- 3.1 Configuration Item Identification
- 3.2 Identification Conventions
- 3.3 Naming Conventions
- 3.4 Labels
- 3.5 Configuration Baseline Management
- 3.6 Libraries

4. Configuration Control
- 4.1 Change Process
- 4.2 Review and Control Board(s)
- 4.3 Interface Management

5. Configuration Status Accounting

6. Configuration Audits

7. Reviews

8. CM Plan Maintenance

4.5 Quality Assurance

QA is an independent function that objectively monitors and reports the application of methodology, policies, processes, procedures, and standards that contract and/or project personnel use to develop software products and services and/or hardware deliverables.

QA provides the methods and procedures to ensure that the software meets its requirements. The principle objective of QA is to assure integrity of the following:

- Deliverable software and its documentation
- Processes used to produce deliverable software

QA's role is to ensure that development systems are compliant with the SDLC and other established IT standards. In addition, QA involvement in project planning activities ensures project quality objectives are well defined, documented, and understood by all; and these quality objectives are supported by the proposed project plans and procedures that govern how tasks will be performed.

QA works with the customer, Test and Evaluation (T&E), and CM to ensure that all necessary SDLC activities are performed. QA conducts audits and reviews at planned milestones and on special occurrences of unplanned events to determine compliance to project processes, to standard organizational processes, and to customer-defined standards. The format for creating a QA plan is contained in Exhibit 9, Quality Assurance Plan.

Exhibit 9: Quality Assurance Plan Outline

QUALITY ASSURANCE PLAN

Cover Page

Table of Contents

1. General
- 1.1 Purpose
- 1.2 Reference
- 1.3 Objective
- 1.4 Glossary

2. NIH Organization
- 2.1 NIH Customer
- 2.2 System Development
- 2.3 Test and Evaluation
- 2.4 Configuration Management
- 2.5 QA Roles and Responsibilities

3. Processes
- 3.1 General
- 3.2 Process 1
- 3.3 Process 2

4. Problem Reporting and Corrective Action
- 4.1 Quality Action Reports
- 4.2 QA Escalation Procedure
- 4.3 QA Report Formats

5. Tools, Techniques, and Methodologies
- 5.1 SDLC
- 5.2 Policies
- 5.3 Standards
- 5.4 Tools

4.6 RISK MANAGEMENT

Risk management (RM) refers to the assessment of potential outcomes of a project and the likelihood that one or more unsuccessful project outcomes may result. It also refers to the process of accepting, transferring, or mitigating risk. An outline of the RM plan is in Exhibit 10, Risk Management Plan.

The RM plan consists of documenting and identifying project risks; the analysis, assessment, and prioritization of those project risks; laying out plans to implement actions to reduce the project risks throughout the project's life cycle. The plan provides a control mechanism to monitor, report, and direct all risk mitigation activities. The plan will be used by the Project Manager and coordinated with all externally affected project participants.

The RM plan is initiated during the System Concept Development phase and is updated and revised during the subsequent phases up to, and including, the Test phase.

Purpose

In this section, present a clear, concise statement of the purpose of the RM plan. Include the name and code name of the project, the name(s) of the associated system(s), and the identity of the organization that is responsible for writing and maintaining the RM plan.

The tracking of risks in a risk identification list is a critical facet of successful system development management. The risk identification list is used from the beginning of the project and is a major source of input for the risk assessment activity. Following are examples of categories that may be a source of risk for a system development:

- The complexity, difficulty, feasibility, novelty, verifiability, and volatility of the system requirements
- The correctness, integrity, maintainability, performance, reliability, security, testability, and usability of the Systems Development Life Cycle (SDLC) deliverables
- The developmental model, formality, manageability, measurability, quality, and traceability of the processes used to satisfy the customer requirements
- The communication, cooperation, domain knowledge, experience, technical knowledge, and training of the personnel associated with technical and support work on the project
- The budget, external constraints, politics, resources, and schedule of the external system environment
- The capacity, documentation, familiarity, robustness, tool support, and usability of the methods, tools, and supporting equipment that will be used in the system development

Once the risks have been identified, document them in this section as the risk identification list. Steps for developing the risk identification list are the following:

- Number each risk using sequential numbers or other identifiers.
- Identify the SDLC document in which the risk is applicable. For instance, if you are working on the Configuration Management (CM) plan and discover a CM risk, identify the CM plan as the related document.
- Describe the risk in enough detail that a third party who is unfamiliar with the project can understand the content and nature of the risk.

Use the risk identification list throughout the life-cycle phases to ensure that all risks are properly documented.

The project plan and the risk identification list are inputs to the risk assessment. Categorize the risks as internal or external risks. Internal risks are those that you can control. External risks are events over which you have no direct control. Examples of internal risks are project assumptions that may be invalid and organizational risks. Examples of external risks are Government regulations and supplier performance.

Evaluate the identified risks in terms of probability and impact. For each risk item, determine the probability that this will occur and the resulting impact if it does occur.

Use an evaluation tool to score each risk, such as the following:

- 0 No known risk exposure exists.
- 1 Risks are inconveniences without serious impact.
- 2 Risks threaten minor impact to process or product.
- 3 Risks may disrupt the process or degrade the product.
- 4 Serious risk to a major part of the project exists.
- 5 Risk exposure threatens failure of the project.

In this section, document the results of the risk assessment. The risk identification list should be divided into two sections: external risks and internal risks. Then each risk should be scored with an evaluation tool category such as 0 for “no risk,” up to 5 for “very serious risk.”

Review the risk items with high rankings and determine if the risks will be accepted, transferred, or mitigated. With the acceptance approach, no effort is made to avoid the risk item. This approach is usually employed because the risk items are the result of external factors over which you have no direct control. Two types of action are usually taken under the acceptance approach: contingency planning and no action.

You can plan contingencies in case the risk does occur. Thus, the project team has a backup plan to minimize the affects of the risk event. Or you can take no action and accept responsibility if the risk event does indeed occur.

With the transfer approach, the objective is to reduce risk by transferring it to another entity that can better bear it. Two methods of transferring risk are the use of insurance and the alignment of responsibility and authority.
With the mitigation approach, emphasis is on actually avoiding, preventing, or reducing the risk. Some risks can be avoided by reducing the number of requirements or defining them more

completely. For example, careful definition of the scope of a project in an SOW can avoid the possible consequence of “scope creep,” or indecisive, protracted, and uncertain scope objectives.

In this section, identify and describe in detail the actions that will be taken to transfer or mitigate risks that are prioritized as high in Section 4. These actions should ultimately result in the reduction of project risk and should directly affect the project plan and the metrics used for the project.

Activities for reducing the effects of risk will require effort, resources, and time just like other project activities. The actions need to be incorporated into the budget, schedule, and other project plan components. Update the project plan components to ensure the planning and execution of risk action activities. Also, refer to contingency plan documents for any contingency plans that have been identified with the risk acceptance approach.

Risk action plans will be used to direct all risk mitigation activities. The RM plan will need to be monitored and updated throughout the life-cycle phases.

Exhibit 10: Risk Management Plan Outline

RISK MANAGEMENT PLAN

Cover Page

Table of Contents

1. Introduction
- 1.1 Purpose
- 1.2 Background
- 1.3 Scope
- 1.4 Policy
- 1.5 Reference Documents
- 1.6 Glossary

2. Risk Identification List

3. Risk Assessment

4. Risk Prioritization

5. Risk Action Plan

4.7 Testing

The goal of testing is to confirm that both individual system modules and the entire system are executed in accordance with the functional requirements and technical specifications. The types of test activities discussed in the subsequent sections are identified more specifically in the Test phase of the life cycle and are included in the test plan and test analysis report.

- **Unit/Module Testing—**Unit/module testing is performed in the Development phase by the system development team. It consists of module/program level testing by validating the module's logic, adherence to functional requirements, and adherence to technical specifications.
- **Subsystem Integration Testing**—Subsystem integration testing is conducted in the Test phase by the system development team. It tests the system's integrated groupings of software units and modules, otherwise known as subsystems.
- **Test and Evaluation**—Activities validate the correct implementation of functional requirements as defined by the Functional Requirements Document (FRD) and its requirements traceability matrix. T&E participation during all major phases of the development life cycle, beginning with system planning and continuing through the operations and maintenance phase, ensures standardized identification, refinement, and traceability of the requirements as such requirements are allocated to the system components. The primary objectives of T&E include the following:
  - Ensuring that requirements are defined in a manner that is verifiable
  - Supporting the traceability of requirements from the source documentation to the design documentation to the test documentation
  - Verifying the proper implementation of the functional requirements
- **User Acceptance Testing**—User acceptance testing is conducted by the user, who tests every system feature for correctness and conformance to requirements. This test should be conducted using the proposed production platform. System interoperability, all documentation, system reliability, and the degree to which the system meets user requirements will be evaluated. Recovery and restart procedures should also be evaluated and tested.
- **Security Testing and Evaluation (ST&E)**—ST&E is performed in the operational production environment. ST&Es are based on the results of the security risk assessment. ST&Es evaluate compliance with security and data integrity guidelines, and address security backup, recovery, and audit trails. ST&E ensures that all security measures have been properly implemented in the operating environment and are effective for satisfying security requirements. It addresses all aspects of security, including internal controls; hardware, software, and communications security controls; physical and environmental security; and administrative procedural security requirements.

4.8 Cost-Benefit Analysis (CBA)

A CBA is a vital management tool for linking function and budget. The analysis is used to support investments in information technology. A preliminary assessment of costs and benefits is made during the strategic planning process and is refined and updated as appropriate throughout the remainder of the life cycle. At each phase, information is gathered and decisions are made that enable the project team to make increasingly accurate projections of the total costs and benefits of the system over its projected life. The CBA information is used to determine both the establishment and the continuation of a project.

When the information system provides services to the public, NIH managers should quantify the performance of the information system through systematic measurement of outputs. Estimates of benefits are based on improvements to the overall work processes enabled by the new automated system. Benefits need not be restricted to savings in staff years, but can include enhancements to service to customers and other stakeholders, reductions in costs, or improvements in overall quality of the products and/or services produced.

In conducting CBAs to support ongoing management oversight, agencies should seek to maximize return on investment over the information systems life cycle by establishing and evaluating systematic performance measures. These performance measures should include the following:

- Effectiveness of program, product, or service delivery (e.g., quality, speed, cost, customer satisfaction)
- Efficiency of program, product, or service management
- Reduction in burden, including information collection imposed on the public

The CBA must be updated as appropriate throughout the information systems life cycle, and the level of detail should be commensurate with the size of the investment. The revised CBA at each phase of the SDLC provides up-to-date information to ensure the continued viability of an information system before and during implementation. Reasons for updating a CBA may include such factors as significant changes in projected costs and benefits; major changes in requirements, including legislative or regulatory changes; or empirical data based on performance measurement gained through prototype or pilot experience.

4.9 Data Management

Because of the large volumes of data handled by some NIH systems and the increasing trend toward sharing data across systems and programs, an integrated data management approach is required when developing information systems. To ensure effective management of all NIH data, all systems are created and maintained in accordance with best practice data management policies and practices. Life-cycle activities are to be carried out consistently with the existing and planned data management environment and data management concerns will be addressed during all phases of the life cycle.

Every information system project must be cognizant of the NIH Enterprise Architecture model, must use NIH common data repositories (when appropriate), and must develop data access strategies in accordance with the NIH enterprise data management plan. Because certain kinds

of data sharing will become widespread, a rigorous and formal data management approach shall be used. This approach minimizes unexpected, adverse effects on the system and the programs it may support. The objectives of this approach are as follows:

- Ensure that data collected and disseminated meet programmatic requirements fully, including requirements for accuracy and timeliness
- Improve management decision making by providing better access to more accurate and timely data
- Increase productivity in the information collection and processing activities as the understanding and use of available data increases
- Ensure that existing data can be shared to the maximum practicable extent, avoiding the cost and confusion of redundant data collection and storage
- Reduce the cost of system maintenance and the time needed to modify implemented systems by designing more stable and flexible databases

Several important aspects of data management include the following:

- **Data Dictionary**—Use of a data dictionary is mandatory for every system. Data dictionary entries must be prepared for every system to communicate clearly the attributes of the data processed by the system-to-system users and other individuals with an interest in the system data.
- **Data Administration**—For each project, data administration focuses on the relationship of the project to other projects and systems that process shared data. Data administration addresses data definitions, data standards, and mechanisms to ensure consistency of data across systems, data quality control procedures, and related issues that frequently cut across project and system boundaries.

4.10 Privacy Act Considerations

Various Federal statutes and regulations impose specific record keeping requirements related to the development, implementation, and maintenance of information systems. These requirements affect the Requirements Definition phase of the SDLC as follows:

- **Development of Records Disposition Schedule**—Federal regulations require that all records no longer needed for the conduct of the regular business of the agency be disposed of, retired, or preserved in a manner consistent with official Records Disposition Schedules. The decisions concerning the disposition criteria, including when and how records are to be disposed, and coordination with Records Management representatives to prepare the Records Disposition Schedule for the proposed system, are the responsibilities of the new system's proponent organization. It is important, however, for the NIH project manager or contact and system technical leaders to be involved in this process.
  - The project manager should be aware of any programmatic decisions (including records management-related factors) concerning a system for which NIH has development and operational responsibility.

- The project manager (and system technical leaders) may have technical knowledge that could influence the decisions to be made by the proponent organization.

Therefore, the NIH project manager or contact should assist the proponent organization in coordinating with the Records Management representatives to prepare the Records Disposition Schedule for the proposed system. The Records Disposition Schedule is a required deliverable for the Requirements Definition phase of system development.

- **Privacy Act Requirements**—For any system that has been determined to be an official System of Records (in terms of the criteria established by the Privacy Act), a special System of Records Notice must be published in the Federal Register. This Notice identifies the purpose of the system; describes its routine use and what types of information and data are contained in its records; describes where and how the records are located; and identifies who the system manager is. While the Records Management representatives are responsible for determining if a system is a Privacy Act System of Records, it is the system owner or proponent organization's responsibility to prepare the actual Notice for publication in the Federal Register. As with the Records Disposition Schedule, however, it is the NIH project manager's responsibility to coordinate with and assist the Privacy Act Notice.

The System of Records Notice is a required deliverable for the Requirements Definition phase of system development.

4.11 Computer and Telecommunications Security Policy Considerations

NIH's Information Systems Security office is responsible for protecting information assets against loss, theft, damage, and unauthorized destruction, modification, and access. To ensure system security, various security risk management activities must be performed throughout the systems life cycle. Security risk management includes identifying sensitive systems; conducting security risk analysis/assessment; developing sensitive system security plans, contingency plans, and security operating procedures; system security testing; and certifying and accrediting a system for operation in a particular security mode.

To protect NIH's numerous and valuable system resources, effective security policies and procedures must be developed and implemented throughout the NIH. The NIH has security policies and procedures in place and continues to make additions and updates. The primary purpose of these policies and procedures is to provide a level of security commensurate with the value of the asset being protected. These security policies and procedures include administrative, computer and telecommunications information, personnel, and physical security requirements.

The project manager needs to ensure that all security requirements are incorporated in each phase of the life cycle. The project plan, including the work breakdown structure, should identify: how and when the security requirements will be identified, defined, and refined; the security safeguards development efforts required during the project's Design phase; the testing criteria, development, and implementation efforts required during the Design and Testing phases; and the certification and accreditation efforts required during the acceptance testing.

Various security program guidance documents have been written and published to consolidate all existing guidance on this subject and to provide supportive details to describe how security requirements may be defined and how safeguards that satisfy those requirements may be developed and implemented. The supportive details contain policies, procedures, and references that provide authority and specific guidance to be used in the preparation of prescribed documents.

5. Life-Cycle Reviews

The life-cycle review process consists of a series of reviews, shown in Exhibit 11, SDLC Reviews and Tests, conducted in each phase to ensure the successful completion of each phase of the project. This process ensures that all products created during the life cycle meet functional and performance requirements as outlined in all requirements documentation. The requirements for holding specific reviews are determined by the system size and complexity and by management direction.

The completion of a phase represents a logical point at which reviews should occur. The purposes of reviews are as follows:

- To ensure that project direction and goals remain consistent with the organization's strategic plan and goals
- To provide an opportunity to terminate projects that fail to demonstrate an adequate return on investment
- To measure the ongoing progress (that is, budget, schedule, and deliverables) or value added and identify potential problems for corrective actions
- To approve phase results and authorize further work

The purpose of each review must be kept in mind during the review process to achieve each purpose before terminating the review. The appropriate personnel selected for the review team should be described in the project plan. In general, senior management involvement is greatest in the earlier phases of systems development efforts. This involvement declines in the later, more technical phases or in less critical projects involving fewer resources.

Exhibit 11: SDLC Reviews and Tests

Phase	Type of Review	Type of Test
System Concept Development Phase	Concept Development Phase review	
Planning Phase	Planning review	
Requirements Definition Phase	Functional Requirements review	
Design Phase	Software Requirements review Preliminary Design review Final Design review	
Development Phase	Development Phase review	Unit/module test
Test Phase	Test Phase review	Subsystem integration test System test User acceptance test Security test and evaluation
Implementation Phase	Implementation Phase review	
Operations and Maintenance Phase	Postimplementation review Periodic System review	
Disposition Phase	Disposition Phase review	

6. Planning, Methodologies, and Tools

To the extent that it is practical, all systems created and maintained by the NIH should use state-of-the-art development methodologies and modern systems development and maintenance tools. Methodologies and tools must be identified early in the life cycle to ensure consistency across life-cycle activities. Methodologies and tools cannot replace the application of systems life-cycle management. In addition to the life cycle phases described additional methodologies and tools may be used before or during systems development.

6.1 Strategic Planning

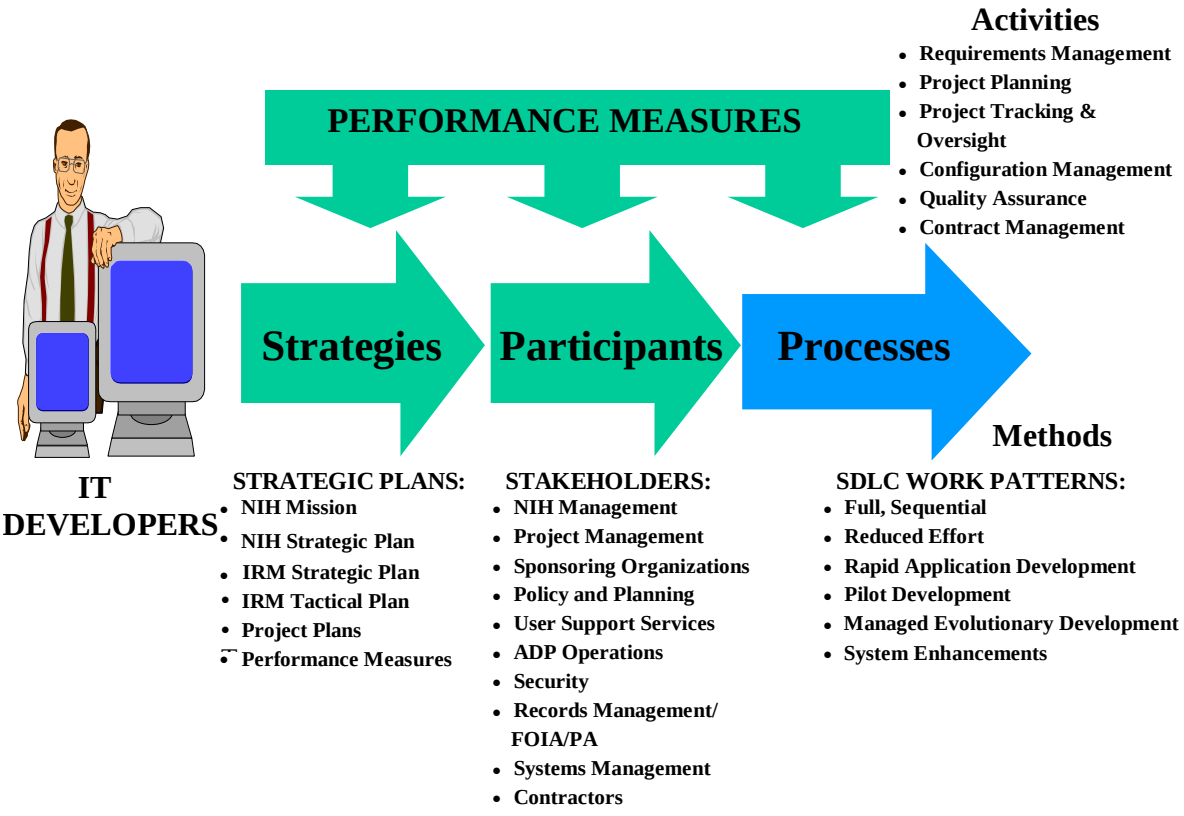
The purpose of Strategic Planning for Information Systems is to ensure that all information systems development activities support the NIH's strategic goals. For these to be known and agreed to throughout the different component agencies, the NIH goes through a business planning process called Strategic Planning. The scope of Strategic Planning includes the entire set of activities that the NIH performs. This planning process produces and updates a plan that is enterprisewide. The plan is also dynamic, meaning that it is always subject to change and to

rigorous change management control. Strategic Planning is not part of the SDLC, but determines what information systems projects get started and/or continue to receive funding. Strategic Planning for Information Systems controls all SDLC application systems projects decided by the benefits each project produces and how these benefits support the NIH Strategic Plan. The description of the Strategic Planning process is outside the scope of the SDLC. However, several important follow-on activities that must be taken are outlined in the next section.

6.1.1 Strategic Management Process

The aim of the Strategic Management Process is to identify potential improvements to NIH information systems and to gain commitment of the required resources to change these systems. Exhibit 12, Strategic Management Process, depicts the major elements of this effort.

Exhibit 12: Strategic Management Process



This strategic management process ensures that effective plans are deployed and that "return on investment" is a key measure of performance. The feedback within the planning process allows for more efficient deployment of Information Resource Management (IRM) services that produce a greater value for the NIH. The planning process communicates to all involved what, how, and when events are to take place, and why they are important. It enables each individual application systems project to develop detailed plans that support the overall NIH effort, while solving project-specific problems.

6.1.2 Performance Measurement

Performance measurement is an essential element in developing effective systems through a strategic management process. The mission, goals, and objectives of the NIH are identified in its strategic plan. Strategies are developed to identify how the NIH can achieve the goals. For each goal, the NIH establishes a set of performance measures. These measures enable the NIH to assess how effective each of its projects is in improving NIH operations.

For the NIH to make this assessment, it determines the current performance level for each measure (performance level baseline) for the existing systems. For each project plan, as part of the economic analysis, the NIH estimates the performance levels it expects to reach as a result of the planned improvements. As the project's improvements are implemented, actual results are compared with the estimated gains to determine the success of the effort. Further analysis of the results may suggest additional improvement opportunities.

The Administration and Congress are seeking improved methods for increasing efficiency in Government. Limited budget resources put an increased emphasis on performance measurement systems. A major step toward measuring results was the passage of the Chief Financial Officer Act of 1990, which requires agencies to provide annual audited reports that emphasize financial and program performance measures. Other laws calling for performance measures are the 1993 Government Performance and Results Act (GPRA) and the 1996 Cohen-Clinger Act (formally IMTRA).

Office of Management and Budget (OMB) Circular A-94, Benefit-Cost Analysis of Federal Programs: Guidelines and Discounts, requires a program-based approach for justifying and evaluating information systems. A program approach requires examining an information system, typically a project, in the context of the functional programs it supports, such as economic analysis, federal budget preparation and monitoring, and drug control policy. It ties the life-cycle cost of the information system to the functional program through the CBA. It ties benefit analysis of the information system and the functional program to measurable high-level business objectives and requirements.

The NIH, using the Strategic Management Process and the appropriate performance measures, defines the functional program that controls and determines which information systems are developed. This strategic management process ensures that the following elements are in place for establishing information system requirements:

- High-level objectives and requirements in the NIH strategic plan that drive the analysis and assessment of feasible alternatives
- Technical requirements clearly linked to the NIH program requirements through the identified performance measures
- Benefits and costs that are identified, analyzed, and measured within the confines of an information system

Performance measures determine whether goals are achieved or requirements are met. They also make possible compliance with OMB Circular A-94, which states the following guidelines:

- Plan for periodic, results-oriented evaluations of program effectiveness using the quantified measures developed in the economic analysis
- Place a high priority on information system projects whose benefits accrue to the general public or to other levels of Government
- Understand that a request for funding approval of most information system projects is based on a reasoned tradeoff between using the funds for the information system and using the funds for other program objectives

6.2 Business Process Reengineering

Business process reengineering (BPR) is performed to change the way an organization conducts its business. BPR is the redesign of the organization, culture, and business processes—sometimes using technology as an enabler—to achieve significant improvements in costs, time, service, and quality. BPR efforts are generally initiated by the strategic planning process. Consequently, these efforts complement and/or augment the Strategic Management Process, and may result in the initiation of an application systems project(s).

- Pre-Select –
  - Establish project Need Based on Strategic Drivers, Goals, Objectives, Initiatives
  - SDLC project created through Initial Investment Decision
- Concept Development –
  - Establish project Context, Concept of Operations, and Implementation Strategy through creation of SDLC EA Products
  - Concurrence with project Concept via Concept Investment Decision
- Business Capability Definition –
  - Define project Business Processes, Data, Performance, Standards, and Initial Infrastructure through creation of SDLC EA Products
  - project Execution Readiness via Business Capability Investment Decision
- Program Execution –
  - Develop & Test project Infrastructure to provide desired Business Capabilities. Demonstrate alignment with EA through creation of defined EA Products.
  - Proceed to Implementation via Implementation Readiness Review
- Implementation –
  - Implement project Infrastructure to provide desired Business Capabilities. Demonstrate alignment with EA through creation of defined EA Products.
  - Proceed to Operations via Operations Readiness Review
- Operations & Maintenance –
  - Support the deployed Business Capabilities and the enabling Infrastructure. Demonstrate alignment with EA through creation of defined EA Products.

7. Mapping to DHHS SDLC

Applications being developed at NIH are being asked to map to the HHS SDLC for budget purposes. Following is a map of both the waterfall and iterative SDLC Framework approach to the DHHS SDLC phases.

DHHS SDLC	Waterfall	Iterative
Pre-Select – Establish project Need Based on Strategic Drivers, Goals, Objectives, Initiatives – SDLC project created through Initial Investment Decision	System Concept Development	Initiate
Concept Development – Establish project Context, Concept of Operations, and Implementation Strategy through creation of SDLC EA Products – Concurrence with project Concept via Concept Investment Decision	System Concept Development Planning Phase	Initiate
Business Capability Definition – Define project Business Processes, Data, Performance, Standards, and Initial Infrastructure through creation of SDLC EA Products – project Execution Readiness via Business Capability Investment Decision	Requirements Definition Phase	Build
Program Execution – Develop & Test project Infrastructure to provide desired Business Capabilities. Demonstrate alignment with EA through creation of defined EA Products. – Proceed to Implementation via Implementation Readiness Review	Development Phase Test Phase	Build

DHHS SDLC	Waterfall	Iterative
Operations & Maintenance – Support the deployed Business Capabilities and the enabling Infrastructure. Demonstrate alignment with EA through creation of defined EA Products.	Operations & Maintenance Phase Disposition Phase	Implement

8. Acknowledgements

This document is extensively based on the SDLC issued by the EOP in 1998.

9. Security Considerations

Security issues are not discussed in this memo.

10. Changes

Version	Date	Change	Authority	Author of Change
0.0	January 6, 2004	Original Document.	N/A	Rich McKay
0.1	May 3, 2005	• Changed abstract, status of the memo, and background sections to alter scope with intent of specifying EA requirements but otherwise • Changed to a BCP for organizations without a document SDLC. • Added change log.	Rich McKay	Steve Thornton

Version	Date	Change	Authority	Author of Change
0.2	August 24, 2005	• Added change log. In section 1, added note about iterative type of SDLC, added paragraph on outside drivers, modified Figure 1 to include outside drivers, and added Figure 2, EA Impact on Iterative Systems Development. • In section 2, added note about iterative type of SDLC. • Added section and exhibit on Risk Management after section 4.4, renumbered follow on sections and exhibits accordingly. • Added 4.2 Project Planning to Table of Contents	Rich McKay	Rich McKay

Version	Date	Change	Authority	Author of Change
0.3	November 2, 2005	• Added 7. Mapping to DHHS SDLC • Changed 7. Acknowledgements to 8. Acknowledgements • Changed 8. Security Considerations to 9. Security Considerations • Changed 9. Changes to 10. Changes • Changed 10. Author's Address to 11. Author's Address	Rich McKay	Rich McKay
0.4	December 12, 2005	• Updated Figure 1 image to make more legible. • Updated title and headers.	Rich McKay	Rich McKay and Steve Thornton
1.0	January 31, 2006	• Updated version number and added ARB approval date. • Updated website URL (page 4).	Architecture Review Board (1/25/2006 meeting)	Steve Thornton

11. Author's Address

Richard McKay
6701 Rockledge Dr. MSC7740
Bethesda, MD

Phone: 301-435-0967
Email: mckayr@csr.nih.gov