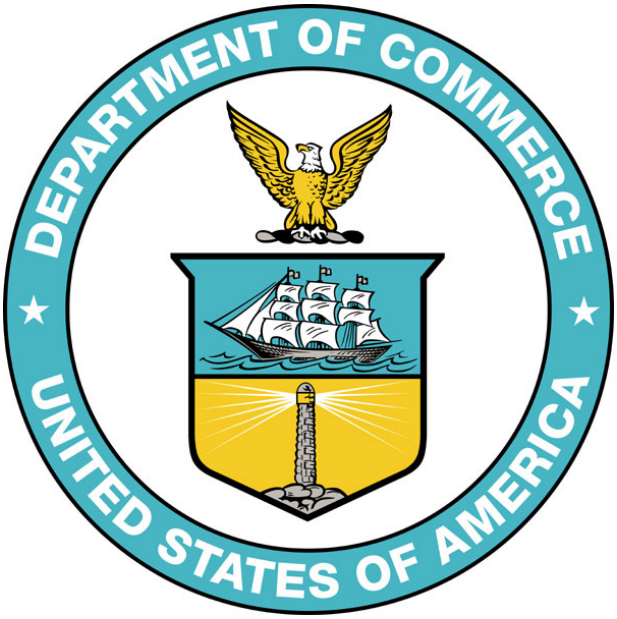


United States Department of Commerce

Enterprise Architecture Program Support

System Development Life Cycle Guide

Version: 2.0

May 30, 2006

Table of Contents

CHANGE LOG

Version	Date	Description
1.0	02/07/2006	Initial Draft
1.1	02/10/2006	Incorporated EAAG Edits
1.2	02/15/2006	Final Edits for Initial Submission to OMB
2.0	08/18/2006	Revised to incorporate OMB comments in Q2 FY06 EA Assessment

Approvals

This System Development Lifecycle Guide has been presented to and approved by:

	Date	Signature
Chief Enterprise Architect		
Director, Office of Information Technology Policy, Planning & Architecture		
DOC Chief Information Officer		

1. PURPOSE OF THIS GUIDE

This Department of Commerce (DOC) System Development Life Cycle Guide provides DOC operating units with an overview of the Department's Information Technology (IT) System Development Life Cycle (SDLC) process. It also outlines the development process that integrates all system engineering activities to produce correct, consistent products effectively and efficiently.

Icon Key

 Stage Gate

 Convergence with Governance Processes

 Action or Process

 Update Deliverable

 Baseline/Re-baseline/Archive Deliverable

Using the Icons

The picture icons at left are provided throughout the guide to help distinguish deliverables and stage gate signoff processes.

They also reveal the convergence of the System Development Life Cycle (SDLC) with other DOC processes including governance processes and the project management methodology.

This guide is also designed to help you comply with DOC policy and execute widely-accepted industry best practices in the development and delivery of DOC IT projects. All of these tools are designed to help you achieve success in development and deployment of software and related components to satisfy project requirements.

After reviewing this guide and by using it to develop and deploy software and related components, you should be able to:

- Understand the activities and tasks within the System Development Life Cycle
- Understand the purpose of and the relationships between each of the DOC IT SDLC processes
- Understand how the DOC IT SDLC process converges with the various DOC IT governance processes
- Identify the deliverables and stage gates required for compliance with DOC policy and standards.

2. CONVERGENCE WITH GOVERNANCE PROCESSES

Within DOC, a number of governance processes guide the development, implementation, and management of the Department's IT resources. The approval and chartering of an IT project requires conformance with the Capital Planning and Investment Control (CPIC) process, the first governance process which serves as a stage gate into the SDLC process. Prior to issuance of a charter, the project must also demonstrate adherence to DOC's Enterprise Architecture (EA) Business Reference

Model (BRM), Once projects are chartered, SDLC convergence with governance processes continues to ensure that architecture, infrastructure, and operational standards are aligned with DOC's strategic IT planning process. Additionally, each operating unit will have internal governance processes that flow into the DOC enterprise governance process. The System Development Life Cycle converges with these governance processes in the Design and Deployment phases. More specifically, review of the System Design Description document is a key milestone in the EA stage gate process and formalized Certification and Accreditation is a requirement for the Approval to Deploy stage gate. Convergence of the SDLC tasks with the various governance processes is indicated in this guide by the 🕮 icon in the Convergence column of each Activity Task table.

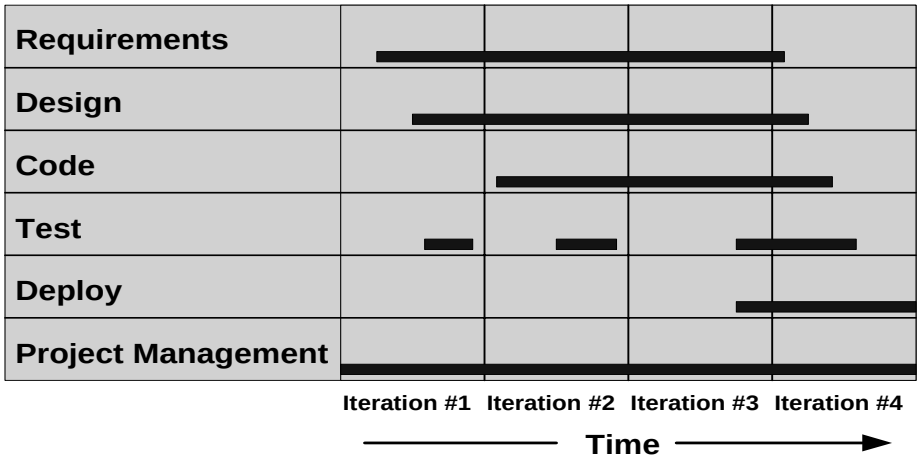
In addition to convergence with the DOC IT governance processes, the SDLC must be closely integrated with the operating unit's project management methodology. Project managers are ultimately responsible for the success of a project and will use the inputs, processes, outputs, and documentation from the SDLC to ensure that success.

3. SYSTEM DEVELOPMENT LIFE CYCLE ACTIVITIES

The following sections provide details on each SDLC activity; a description of the purpose of the activity, entry and exit criteria, and the sequence of tasks to be completed or initiated within each activity.

Various SDLC methodologies have been developed over the years to guide the processes involved, including the waterfall model (which was the original SDLC method), rapid application development (RAD), joint application development (JAD), the iterative development model, the spiral model, and others. Service Oriented Architecture (SOA) developmental processes align themselves closely, but not exclusively, to iterative development methodologies. At a very high level, the Requirements, Design, Coding, and Testing phases would each be a part, in varying degrees, of an iteration.

In the example below, Iteration #1 would include requirements gathering, design work, and perhaps some testing of the Business Process Models developed. Iteration #2 would include more of the above but also include some coding based on the requirements and design work already done.



It is important to note that although this guide is laid out in a linear and sequential format it does not advocate the exclusive use of a waterfall development model. For clarity, the text for each stage may be written as though the stage is done once with a single entry and exit point; however, as seen in the figure above, additional iterations of stages are likely. The documents discussed below make up the necessary artifacts for any good SDLC development model. It is the responsibility of project managers and project architects to choose the proper development model for their project and then tailor the SDLC steps and documents appropriately.

3.1 DOCUMENT SYSTEM REQUIREMENTS

Purpose: To establish a common understanding of the requirements between the customer, stakeholders, and the project team. System requirements are elicited from the customer and other stakeholders, analyzed, and presented in a Systems Requirements Specification (SRS). The SRS is used as the basis for the design and implementation of the system. It is imperative that the SRS be verified with the customer and stakeholders to ensure that the system requirements are correct and complete and that they are accurately represented within the Project Concept Proposal, Project Business Case, Project Charter, and Project Plan. The SRS must also demonstrate adherence to the DOC EA Service Component Reference Model (SRM) and be approved by the EA Review Board.

Entry Criteria:

- Project Plan Stage Gate Signoff
- Project Concept Proposal, Project Business Case, Project Charter (as appropriate), and Project Plan are complete and approved by the project sponsor
- Project scope is elaborated and documented in the project plan

Inputs:

- Project Concept Proposal
- Project Business Case
- Project Charter
- Project Plan

- EA Business Reference Model
- Existing SRS, Requirements Traceability Matrix (RTM), if existing from previous projects
- Existing software baseline(s), as appropriate
- Test Plan, if existing from previous projects
- Acceptance Test Scenarios, if existing from previous projects

Activity Tasks: Document System Requirements

Task No.	Deliverable Stage Gate	Convergence	Task
3.1.1	☐		Elicit and define system requirements. Convergence with Select phase of CPIC process
3.1.2	☐		Construct SRS including Business Process Models, Use Cases, Use Case Diagrams, Web services requirements, and/or storyboards as appropriate.
3.1.3	☐		Conduct Review of SRS.
3.1.4	☑		**Stage Gate: Obtain System Requirements Specification Signoff.** Convergence with EA Business Reference Model
3.1.5	☐		Begin construction of Acceptance Test Phase section of the Test Plan.
3.1.6	☐		Define performance metrics in relation to the Performance Reference Model
3.1.7	☐		Begin construction of Acceptance Test Scenarios.
3.1.8			Begin construction of Deployment Plan.
3.1.9			Capture baselines and place under configuration management. ◦ Baseline SRS ◦ Re-baseline project management deliverables, as appropriate.

Exit Criteria:

- SRS is baselined
- **SRS Stage Gate Signoff**

Outputs:

- SRS baseline – includes all known requirements. Can also include Business Process Models, Use Cases, Use Case Diagrams, Web services requirements, and/or storyboards.
- Re-baseline of project management deliverables, as appropriate
- Test Plan draft
- Acceptance Test Scenarios draft
- Performance Measures
- Deployment Plan draft

3.2 DESIGN SYSTEM

Purpose: To develop a system architecture and design that meet Departmental and operating unit Enterprise Architecture requirements.

Entry Criteria:

- SRS is baselined
- **SRS Stage Gate Signoff**

Inputs:

- EA Service Component Reference Model
- EA Technical Reference Model and Standards Profiles
- SRS baseline - include Business Process Models, Use Cases, Use Case Diagrams, Web services requirements, and/or storyboards
- Existing software baseline(s)
- Departmental and operating unit methodologies and best practices
- Test Plan draft
- Acceptance Test Scenarios draft
- Deployment Plan draft

Activity Tasks: Design System

Task No.	Deliverable Stage Gate	Convergence	Task
3.2.1	☐	🕮	Develop System Design. Convergence with EA Service Component Reference Model

☐

Task No.	Deliverable Stage Gate	Convergence	Task
3.2.2			Construct System Design Description (SDD), including process flow diagrams, process traceability matrix, portal specifications, user interface traceability matrix, services list, services traceability matrix, and Web Services Description Language (WSDL) specifications as appropriate.
3.2.3	☐		Construct RTM.
3.2.4			Update SRS, as appropriate.
3.2.5	☐		Conduct review of SDD.
3.2.6	☑		**Stage Gate: Obtain System Design Description Signoff.** Convergence with EA Service Component Reference Model, and Technical Reference Model
3.2.7	☐		Begin construction of Introduction, Test Methodology, Test Schedule, Test Monitoring and Reporting sections of the Test Plan.
3.2.8	☐		Construct Integration and System Test Phase sections of the Test Plan.
3.2.9	☐		Construct Integration and System Test Scenarios, including traceability to requirements.
3.2.10	☐		Initiate construction of user and support training materials
3.2.11			Update Acceptance Test Phase section of the Test Plan.
3.2.12			Update Acceptance Test scenarios.

Task No.	Deliverable Stage Gate	Convergence	Task
3.2.13			Update Deployment Plan.
3.2.14			Capture baselines and place under configuration management: ◦ Baseline SDD and RTM ◦ Re-baseline SRS, as appropriate

Exit Criteria:

- SDD is baselined
- System design demonstrates adherence to EA Technical Reference Model
- **SDD Stage Gate Signoff**

Outputs:

- SDD and RTM baseline
- Test Plan draft, baselined
- Integration and System Test Scenarios draft, baselined
- Acceptance Test Scenarios draft, baselined
- Deployment Plan draft, baselined
- User Training materials draft, baselined
- SRS re-baseline, as appropriate

3.3 CODE SYSTEM DESIGN

Purpose: To construct the system product described by the SDD.

Entry Criteria:

- SDD is baselined
- **SDD Stage Gate Signoff**

Inputs:

- EA Data Reference Model (DRM)
- SRS, SDD, and RTM baseline
- Existing software baseline(s), as appropriate
- Departmental and operating unit methodologies, standards, and best practices
- Test Plan draft
- Acceptance Test scenarios
- User and support training materials, draft

Activity Tasks: Code System Design

Task No.	Deliverable Stage Gate	Convergence	Task
3.3.1	☐		Analyze SDD.
3.3.2	☐		Analyze existing software baseline(s).
3.3.3	☐		Construct source code. Review and use existing service repositories as appropriate. Convergence with EA Data Reference Model
3.3.4	☐		Create executables.
3.3.5	☐		Conduct Code Reviews.
3.3.6	☐		Finalize Unit Test section of Test Plan
3.3.7	☐		Construct Unit Test Scenarios.
3.3.8			Update Acceptance Test Phase section of the Test Plan.
3.3.9			Update Acceptance Test scenarios.
3.3.10			Update RTM.
3.3.11			Update SRS and SDD, as appropriate.

Task No.	Deliverable Stage Gate	Convergence	Task
3.3.12			Capture baselines and place under configuration management. ◦ Baseline source code and executables ◦ Re-baseline RTM ◦ Re-baseline SRS and SDD, as appropriate ◦ Re-baseline project management deliverables as appropriate

Exit Criteria:

- Executables are ready for Unit Test
- Unit Test Scenarios
- Source code demonstrate adherence to EA Data Reference Model

Outputs:

- Source code and executables
- RTM
- Test Plan draft
- Unit Test Scenarios
- Acceptance Test Scenarios
- SRS and SDD re-baseline, as appropriate

3.4 PERFORM UNIT TEST

Purpose: To ensure that each unit executes as designed.

Entry Criteria:

- Executables are ready for Unit Test
- Unit Test Scenarios completed

Inputs:

- Source code and executables
- SDD and RTM baseline
- Previous test results, as appropriate
- Test environments
- Test Plan draft
- Unit Test Scenarios

Activity Tasks: Perform Unit Test

Task No.	Deliverable Stage Gate	Convergence	Task

Task No.	Deliverable Stage Gate	Convergence	Task
3.4.1			Execute Unit Test. Correct source code/executables and retest (Task 3.4.1) as necessary. Convergence with EA Data Reference Model
3.4.2			Validate test results.
3.4.3			Update Unit Test section of Test Plan, as appropriate.
3.4.4			Update Unit Test Scenarios.
3.4.5			Update user and support training materials as appropriate
3.4.6			Construct Unit Test Report to document test results.
3.4.7			Capture baselines and place under configuration management. ◦ Baseline source code and executables ◦ Baseline Unit Test Report

Exit Criteria:

- Unit Test successful.
- Source code and executables for each unit are ready for Integration Test.
- Unit Test Report completed
- Source code and executables demonstrate adherence to EA Data Reference Model

Outputs:

- Tested source code and executables baseline from Unit Test
- Test Plan draft
- Unit Test Scenarios
- Draft user and support training materials
- Unit Test Report

3.5 PERFORM INTEGRATION TEST

Purpose: To ensure that units tested during unit testing can be successfully integrated with other units.

Entry Criteria:

- Integration Test Scenarios completed
- Source code and executables for each unit are ready for Integration Test

Inputs:

- Tested source code and executables baseline from Unit Test
- SDD and RTM baseline
- Previous test results, as appropriate
- Test Plan draft
- Test data
- Integration Test Scenarios
- Departmental and operating unit methodologies

Activity Tasks: Perform Integration Test

Task No.	Deliverable Stage Gate	Convergence	Task
3.5.1			Conduct Review of Test Plan and Test Scenarios
3.5.2			Capture baselines and place under configuration management. ◦ Baseline Test Plan and Test Scenarios
3.5.3			Prepare Integration test environment, as described in the Test Plan.
3.5.4			Conduct test-readiness review: ◦ Status of code development ◦ status of unit tests ◦ readiness of test environment ◦ availability of resources
3.5.5			Execute Integration test. Correct source code/executables and retest (Task 3.5.5) as necessary. Convergence with EA Data Reference Model and Service Component Reference Model

Task No.	Deliverable Stage Gate	Convergence	Task
3.5.6			Validate test results.
3.5.7			Update Integration Test Phase and System Test Phase sections of Test Plan, as appropriate.
3.5.8			Update Integration Test Scenarios, as appropriate.
3.5.9			Construct Integration Test Report to document test results.
3.5.10			Capture baselines and place under configuration management. ◦ Baseline source code and executables ◦ Baseline Integration Test Report

Exit Criteria:

- Integration Test successful
- Source code and executables are ready for System Test
- Source code and executables demonstrate adherence to EA Data Reference Model
- Integration of unit-level sub-systems demonstrates adherence to EA Service Component Reference Model

Outputs:

- Tested source code and executables baseline from Integration Test
- Test Plan
- Integration Test Scenarios
- Integration Test report

3.6 PERFORM SYSTEM TEST

Purpose: To ensure that integration-tested executables can be integrated into a system.

Entry Criteria:

- Integration Test successful
- Source code and executables are ready for System Test

Inputs:

- Tested source code and executables baseline from Integration Test
- SDD and RTM baseline
- Previous test results, as appropriate
- Test Plan
- Unit and Integration Test data
- System Test Scenarios
- Departmental and operating unit methodologies

Activity Tasks: Perform System Test

Task No.	Deliverable Stage Gate	Convergence	Task
3.6.1			Conduct review of Test Plan and Test Scenarios, as appropriate.
3.6.2			Capture baselines and place under configuration management, as appropriate. ◦ Baseline Test Plan and Test Scenarios.
3.6.3			Prepare System Test environment, as described in the Test Plan.
3.6.4			Conduct test-readiness review: ◦ Status of code development ◦ status of unit tests ◦ readiness of test environment ◦ availability of resources
3.6.5			Execute System Test. Correct source code/executables and retest (Task 3.6.5) as necessary. Convergence with EA Data Reference Model and Service Component Reference Model
3.6.6			Validate test results. Convergence with Performance Reference Model
3.6.7			Update System Test Phase section of Test Plan, as appropriate.
3.6.8			Update System Test Scenarios, as appropriate.

Task No.	Deliverable Stage Gate	Convergence	Task
3.6.9			Construct System Test Report to document test results.
3.6.10			Capture baselines and place under configuration management: ◦ Baseline source code and executables ◦ Baseline Test Plan ◦ Baseline System Test Scenarios ◦ Baseline System Test Report ◦ Project management deliverables, as appropriate

Exit Criteria:

- System Test successful
- Source code and executables are ready for Acceptance Test
- Source Code and executables demonstrate adherence to EA Data Reference Model
- System configuration demonstrates adherence to Service Component Reference Model
- Test results demonstrate adherence to EA Performance Reference Model

Outputs:

- Tested source code and executables baseline from System Test
- Test Plan
- System Test Scenarios
- System Test Report

3.7 PERFORM ACCEPTANCE TEST

Purpose: To perform all Acceptance Test tasks for a given system release.

Entry Criteria:

- System Test successful
- Source code and executables are ready for Acceptance Test

Inputs:

- Tested source code and executables baseline from System Test
- RTM, SDD baseline
- Previous test results, as appropriate
- Test Plan
- Unit, Integration, and System Test data
- Acceptance Test Scenarios
- User and support training materials draft

- Departmental and operating unit methodologies

Activity Tasks: Perform Acceptance Test

Task No.	Deliverable Stage Gate	Convergence	Task
3.7.1	☐		Prepare Acceptance Test environment, - as described in the Test Plan.
3.7.2	☐		Finalize user and support training materials.
3.7.3	☐		Conduct test readiness review, including as a minimum: ° Status of code development ° Status of unit, integration, and system tests ° Readiness of test environment ° Resource availability
3.7.4	☐		Execute Acceptance Test. Correct source code/executables and retest (Task 3.7.4) as necessary.
3.7.5	☐		Validate test results.
3.7.6			Update Acceptance Test Phase section of Test Plan, as appropriate.
3.7.7			Update Acceptance Test Scenarios, as appropriate.
3.7.8			Update user and support training materials.
3.7.9	☐		Construct Acceptance Test Report to document test results.
3.7.10	☐		Obtain customer sign-off of Acceptance Test results. Convergence with EA Performance Reference Model

Task No.	Deliverable Stage Gate	Convergence	Task
3.7.11			Conduct Deployment Readiness Review: ◦ Invite stakeholders ◦ Prepare materials ◦ Distribute materials
3.7.12			Capture baselines and place under configuration management: ◦ Baseline source code and executables ◦ Baseline Test Plan ◦ Baseline Acceptance Test Scenarios ◦ Baseline Acceptance Test Report ◦ Baseline User Training Materials ◦ Update project management deliverables, as appropriate.

Exit Criteria:

- Acceptance Test successful, with customer/sponsor accepting unresolved defects, as appropriate
- Source code and executables are ready for deployment
- System performance requirements adhere to EA Performance Reference Model
- User Training Materials are complete

Outputs:

- Tested and corrected source code and executables baseline from Acceptance Test
- Updated Test Plan
- Updated Acceptance Test Scenarios
- Updated Acceptance Test Report
- Customer sign-off of Acceptance Test results
- Updated User Training Materials

3.8 PREPARE FOR DEPLOYMENT

Purpose: To finalize, review, and initiate execution of the Deployment Plan and to obtain Approval to Deploy.

Entry Criteria:

- Acceptance Test successful, with customer/sponsor accepting unresolved defects, as appropriate

Inputs:

- Deployment Plan draft
- Project management and SDLC artifacts
- Prior Stage Gates Signoff
- Appropriate infrastructure documentation required for production set-up and release
- Departmental and operating unit methodologies
- Source code and executables

Activity Tasks: Prepare for Deployment

Task No.	Deliverable Stage Gate	Convergence	Task
3.8.1	☐		Finalize Deployment Plan.
3.8.3	☐		Begin execution of Deployment Plan, to include as a minimum: ◦ Prepare site (production environment) ◦ Prepare production system release ◦ Execute Configuration Audit ◦ Create/convert data ◦ Prepare appropriate infrastructure documentation required for production set-up and release ◦ Conduct training ◦ Prepare Notification of Deployment including build or release notes developed in a non-technical format for end users of the system
3.8.4	☐		Conduct Deployment Readiness Review:
3.8.5	☐		Ensure completeness of project management deliverables; including as a minimum: ◦ IT Security Plan ◦ Risk Management Plan ◦ Certification and Accreditation documentation

Task No.	Deliverable Stage Gate	Convergence	Task
3.8.6			Update Deployment Plan, as appropriate
3.8.6			**Stage Gate: Obtain Approval to Deploy Signoff.** Convergence with CPIC process, including system Certification and Accreditation
3.8.7			Capture baselines and place under configuration management. ° Baseline Plan ° Baseline Approval to Deploy Document ° Baseline appropriate infrastructure documentation required for production set-up and release ° Project management deliverables, as appropriate.

Exit Criteria:

- Deployment Plan is baselined
- Deployment Readiness Review conducted
- Approval to Deploy document is baselined
- **Approval to Deploy Stage Gate Signoff**

Outputs:

- Deployment Plan baseline
- Approval to Deploy document baseline
- Appropriate infrastructure documentation required for production set-up and release baseline
- User and support training materials and documentation

3.9 DEPLOY

Purpose: To successfully deploy the system into production. Note that all of the tasks in this phase, including planning and execution of system deployment, should be identified in the project manager's detailed schedule.

Entry Criteria:

- Baselined Deployment Plan
- Deployment Readiness Review conducted
- Baselined Approval to Deploy document

- **Approval to Deploy Stage Gate Signoff**

Inputs:

- Deployment Plan baseline
- Appropriate documentation for production release
- User and support training materials
- Departmental and operating unit methodologies
- Source code and executables

Activity Tasks: Deploy

Task No.	Deliverable Stage Gate	Convergence	Task
3.9.1			Complete execution of Deployment Plan, including as a minimum: ° Prepare site (production environment) ° Prepare production system release ° Execute configuration audit ° Create/convert data ° Distribute training materials and documentation ° Release system ° Distribute Notification of Deployment including build or release notes developed in a non-technical format for end users of the system ° Execute Contingency Plan, if necessary
3.9.2			Capture baselines and place under configuration management. ° Baseline production source code and executables ° Baseline Configuration Audit results ° Baseline data creation/conversion source code and executables ° Baseline training materials Convergence with all EA Reference Models Convergence with Evaluate phase of CPIC process

Exit Criteria:

- Successful deployment or successful execution of Contingency Plan
- Production source code and executables are baselined
- Data is created or converted and adheres to the EA Data Reference Model
- Training materials are baselined
- System baselines adhere to EA Technical Reference Model
- Demonstrated system performance adheres to the EA Performance Reference Model
- System adheres to and is documented in the EA Business Reference Model and Service Component Reference Model

Outputs:

- Production environment
- Production source code and executables baseline
- Configuration Audit results baseline
- Data creation/conversion source code and executables baseline
- Created or converted data
- Training materials baseline
- Updates to EA Business Reference Model and Service Component Reference Model, as appropriate
- Notification of Deployment document

4. APPENDICES

4.1 GLOSSARY

Term	Definition
Acceptance testing	Determination of the correctness of all functionality of the system in order to provide the basis for acceptance of the system.
Analyze	The act of examining a complex whole by dividing it into its parts or elements. To analyze suggests separating or distinguishing the component parts of something (as a substance, a process, a situation) so as to discover its true nature or inner relationships.
Baseline	A work product that has been formally reviewed and accepted by the involved parties. A baseline should be changed only through formal configuration management procedures. Some baselines may be project deliverables while others provide the basis for further work.
Business Reference Model	The Business Reference Model is a function-driven framework for describing the business operations of the Department. The BRM is the first layer of the Department's Enterprise Architecture and it is the main viewpoint for the analysis of data, service components and technology.
Certification and Accreditation	A set of procedures and judgments leading to a determination of the suitability, from an information assurance standpoint, of an information technology system to operate in an operational environment.
Correctness	The extent to which an item under test satisfies specifications and fulfills user's mission objectives.
Deliverable	Any measurable, tangible, verifiable outcome, result, or item that must be produced to complete a project or part of a project. Often used more narrowly in reference to an external deliverable; that is, a deliverable that is subject to approval by the project sponsor or customer.
Deploy	To place into appropriate position or arrange strategically.

Term	Definition
Data Reference Model	The Data Reference Model is a framework to promote the common identification, use, and appropriate sharing of data/information across the Department. It provides standards and guidelines to help operating units structure, categorize, exchange, and manage their data to improve the Department's ability to perform Department-wide information sharing.
Enterprise Architecture	The practice of applying a comprehensive and rigorous method for describing a current or future structure for an organization's processes, information systems, personnel and organizational sub-units, so that they align with the organization's core goals and strategic direction.
Integration testing	The determination of the correctness of the aggregate with regard to its associated requirements.
Performance Reference Model	The PRM is a "reference model" or standardized framework to measure the performance of major IT investments and their contribution to program performance.
Project Lifecycle	A collection of generally sequential project phases whose names and number are determined by the control needs of the organization or organizations involved in the project. A life cycle can be documented with a methodology.
Service Component Reference Model	The Service Component Reference Model (SRM) is a business and performance-driven, functional framework that classifies Service Components with respect to how they support business and/or performance objectives.
System Development Life Cycle	A conceptual model used in project management that describes the stages involved in an information system development project. The stages include an initial feasibility study, planning, resource allocation and risk assessment through requirements gathering, design, implementation, integration, testing, deployment, maintenance; and finally retirement of the completed application. These phases may overlap or be performed iteratively.
System testing	The determination of the correctness of the system and verification that the system conforms to stated requirements.
Technical Reference Model	The Technical Reference model is used to describe and align the technical standards, specifications, and technology capabilities supporting the delivery, exchange, and construction of business (or service) components. The TRM provides a foundation to advance the re-use of technology and component services from a Department-wide perspective.

Term	Definition
Test	The methodical (and typically documented) combination of a set of inspections, analyses, and demonstrations that provide a well-defined test procedure to assure the correctness of some functionality in the item under test.
Unit testing	The determination of the correctness of the specific functionality of a module of source code or a component of a larger system.
Verification	The determination that the developmental products of a given activity conform to the stated requirements for those products.
Validation	A determination that the system satisfies the intended use and user needs.

4.2 LIST OF ACRONYMS

Acronym	Meaning
DOC	Department of Commerce
DRM	Data Reference Model
EA	Enterprise Architecture
IT	Information Technology
JAD	Joint Application Development
PRM	Performance Reference Model
RAD	Rapid Application Development
RTM	Requirements Traceability Matrix
SDD	System Design Description
SDLC	System Development Life Cycle
SOA	Service Oriented Architecture
SRM	Service Component Reference Model
SRS	Systems Requirements Specification
WSDL	Web Services Description Language