Oregon Public Employees Retirement System	Posted date December 21, 2006	Number 3.05.01.00.168.POL
Signature *Eric Sokol*	Approval date December 21, 2006	Page 1 of 24
Policy:	Systems Development Life Cycle (SDLC)	
Objective:	Explains the process for SDLC, which provides coordinated service to support PERS staff.	
Reference:	Business Process Modeling Methodology document (see attachment)	

Executive Summary

This document provides an overview of the Systems Development Life Cycle (SDLC) process for the Oregon Public Employees Retirement System (PERS). The SDLC process consists of eight phases that help manage a wide variety of activity to conduct projects or automate PERS' activities with information technology. SDLC is not limited to technical activity but actually begins with customer needs and evolves through processes and user requirements to develop a solution or support process. A detailed explanation of SDLC procedures is outlined in a separate document and provides guidance, templates, checklists, and examples for successful implementation of this policy. The primary objective of implementing a standardized SDLC policy is to provide coordinated excellent service to support the activity of customers and users within PERS.

The first section of the policy explains the purpose, background, and basic systems development concepts in order to establish a context for policy description. The end user development methodology, project management practices, and management controls make up the SDLC environment to which this policy applies. The SDLC phase concept is further explained to ensure the policy or "ground rules" is understandable by individuals other than technology specialists. A simplified and common framework for implementing SDLC will improve communications and promote coordination across projects throughout PERS' community. The eight phases of SDLC are:

- Initiation,
- System Concept Development,
- Planning,
- Requirements Analysis,
- Design,
- Development,
- Integration and Test, and
- Implementation.

The second section of this policy explains selected key terms, describes the SDLC concept, and describes each phase of SDLC in greater detail. SDLC is not a common practice in performing daily tasks but is critical to develop ways to support daily tasks and requires the cooperation of everyone involved. This SDLC policy requires that eight SDLC phases are used to conduct projects within PERS' environment and an electronic project notebook (on K drive) maintained to track the status of a project related to the SDLC phases.

This SDLC describes an overall structured approach to information management. Primary emphasis is placed on the information and systems decisions to be made and the proper timing of decisions. The SDLC provides a flexible framework for approaching a variety of systems projects. The framework enables system developers, project managers, program/account analysts, and system owners/users to combine activities, processes, and products, as appropriate, and to select the tools and methodologies best suited to the unique needs of each project. This complete SDLC will not be used for RCP Projects. For RCP Projects, a combination of PERS' SDLC components (e.g., Project Management, QA) and the vendor's SDLC will be used.

Section 1 — Policy Overview

1.1. Purpose

This policy establishes a consistent set of management practices and terms to conduct systems development in the PERS environment. This policy also establishes a standardized process for conducting projects and automating PERS' business processes.

1.2. Background

PERS is a unique environment requiring the customization of industry technology practices to automate PERS' activity for effective and efficient operations. Tailoring an SDLC policy in the PERS environment allows PERS to benefit from industry technology and avoid costly software development. A clear and simplified SDLC policy provides a basis for PERS' staff participation in systems development in a way that provides common understanding and promotes checklist style steps. A structured and consistent approach to systems development ensures successful technology initiatives that are coordinated among the different areas within PERS.

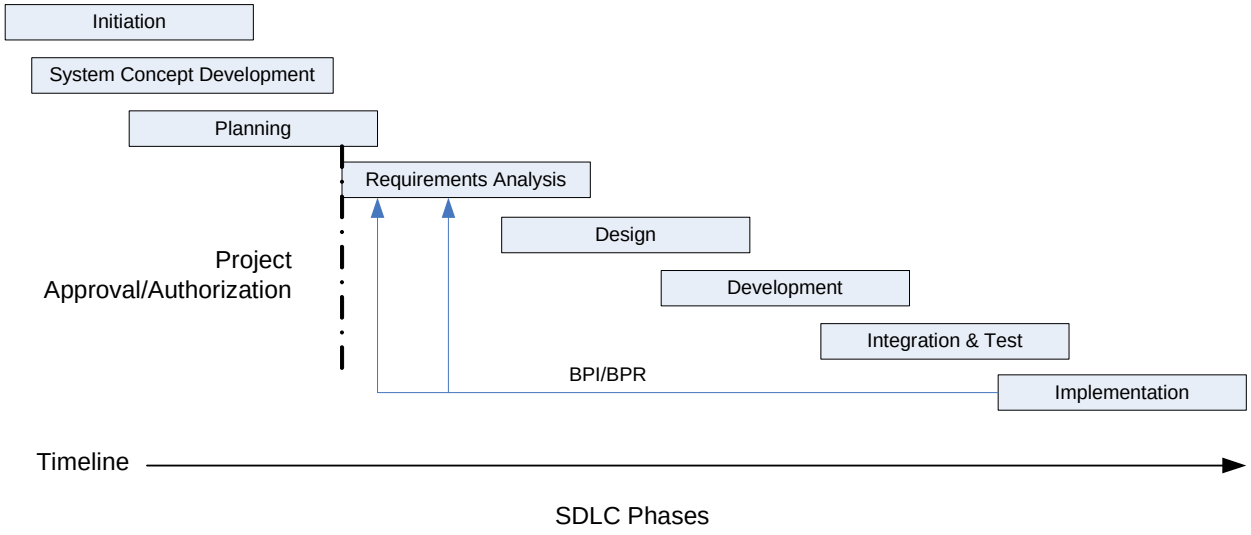
1.3. Systems Development

Systems development in the PERS environment is a specific effort to automate PERS' activity (business processes) by using hardware, software, people, and procedures. The SDLC process is defined as an organized way to determine customer needs and user requirements such that technology

can be applied through systems development, to help customers and users perform their jobs more effectively and efficiently. Project management is a tool used to manage the use of a systems development methodology and ensures systems are built that helps users and customers. The SDLC environment includes an end user development methodology, project management guide, and management controls to establish a level of checks and balances to ensure a successful project.

1.4. SDLC Phase Concept

The SDLC phase concept is used to describe functional systems development activity, to gain control of the complexities of systems development, and to ensure the needs of customers and users are the basis for technical activity. The SDLC process is best described as a series of phases occurring in various degrees and stages of overlap. The diagram below shows the eight major phases of systems development tailored to the PERS environment and the overlap that may occur during execution.



The SDLC phases provide an excellent opportunity to control, monitor, and audit the systems development process and to ensure customer and user satisfaction. A feedback process supporting Business Process Improvement (BPI) and Business Process Reengineering (BPR) is critical to ensure that the correct process or activity is the basis for the technical activity leading to systems engineering and automation.

It is necessary to draw a distinction between the eight SDLC functional activities (phases) and functional tasks that occur within the phases. The SDLC phases provide an opportunity for PERS to manage and track the creation, approval, and progress of projects or information technology

initiatives. The functional tasks are managed for each project and may occur within one or more phases. Specific tasks, such as technology evaluations and demonstrations, may be conducted across several SDLC phases to ensure user and system requirements are defined that are achievable with current technology. It is important to identify the mandatory tasks for all projects to provide consistency among PERS' projects. The eight SDLC phases should be used as a checklist to ensure that each project is managed in a consistent and thorough manner at a level matching the project scope. Functional tasks will be described in section two of this policy as they relate to the SDLC.

1.5. SDLC phases

Summary definitions of the eight SDLC phases are provided below:

Initiation – The initiation phase prompts the collection of information used to determine if a project warrants the investment of personnel and funding. It marks the identification of a business need or opportunity. The initiation phase should identify the customer, user, mandate, basic operating concept, benefits, and impact to the organization. This information is formally captured in the business case document.

System Concept Development – The system concept development phase should identify a preliminary investigation of alternatives and risk analysis and a high-level cost-benefit analysis to determine if the project has favorable return on investment. The project information should show the expected cost increase or decrease and the capabilities or benefits gained. This information is captured in the Project Proposal document. The key output of this phase is knowing exactly what the scope of the project is prior to committing funding and resources and a formalized approval/authorization or disapproval of the project based on the project definition.

Planning – The planning phase is the most critical step in completing development, acquisition, and maintenance projects. Careful planning, particularly in the early stages of a project, is necessary to coordinate activities and manage project risk effectively. Project plans refine the information gathered during the initiation phase by further identifying the specific activities and resources required to complete a project. A critical part of a project manager's job is to coordinate discussions between user, audit, security, design, development, and network personnel to identify and document as many functional, security, and network requirements as possible.

Requirements Analysis – Requirements analysis can be broken down into two parts: user requirements and system/data requirements. The user requirements portion of this phase is based on the processes that users conduct in their day-to-day activities. The user requirements should clearly describe what part of the user process (activity) should be automated or enhanced, and the expected capabilities and features. There may be a number of other tasks to perform prior to developing the user

requirements, such as interviews, objectives identification, and defining operating concepts. The key output of this portion of the phase is a summary document of user requirements that explains what the system is supposed to do. The system/data requirements portion of this phase is based on merging user processes and requirements to allow a system to support many different users or functions in similar areas. The system/data requirements portion of this phase marks the start of establishing key technical areas to use technology to make the work easier. Technical requirements are usually established as a sub-set of the system/data requirements and incorporate existing systems and technologies currently used by the organization. Data requirements are a critical road map for any information system designed to process data. The key output of this phase is a summary document of system/data requirements explaining what the system should be built to, how data should be processed, and what technical or support requirements may exist. The user and system/data requirements will be summarized into a Software Requirements Specification document. In addition, security and internal control-related requirements are also developed as appropriate to the scope of the project.

Design – The design phase is a complex and critical step in determining which system design, based on systems engineering and technology analysis, and meets the user and system/data requirements. For non-technical solutions, the design may simply be a support process to be implemented over time. The design may be presented as several options with trade-off analysis or a specific configuration and may consist of commercial-off-the-shelf (COTS) products (preferred approach) or customized development. Procurement options and cost information should be identified as determined by resource requirements and the design. The most significant milestone in this phase is the recommendation of what to do or buy to meet the user and system/data requirements. The development of appropriate documentation, such as the Conversion Plan, Implementation Plan, Maintenance Manual, *Operations Manual*, Training Plan, and *User Manual*, begins in this phase to ensure successful transition during implementation.

Development – The development phase is the execution of the approved design and in some cases may blend into the implementation phase. A smaller test system is sometimes a good idea to get a proof-of-concept validation before committing funds for large scale fielding of a system without knowing if it really works as intended by the user. For non-technical solutions, the development may involve creation of a support process and move directly to implementation. Procurement activity begins in this phase and may be expanded with deployment during implementation.

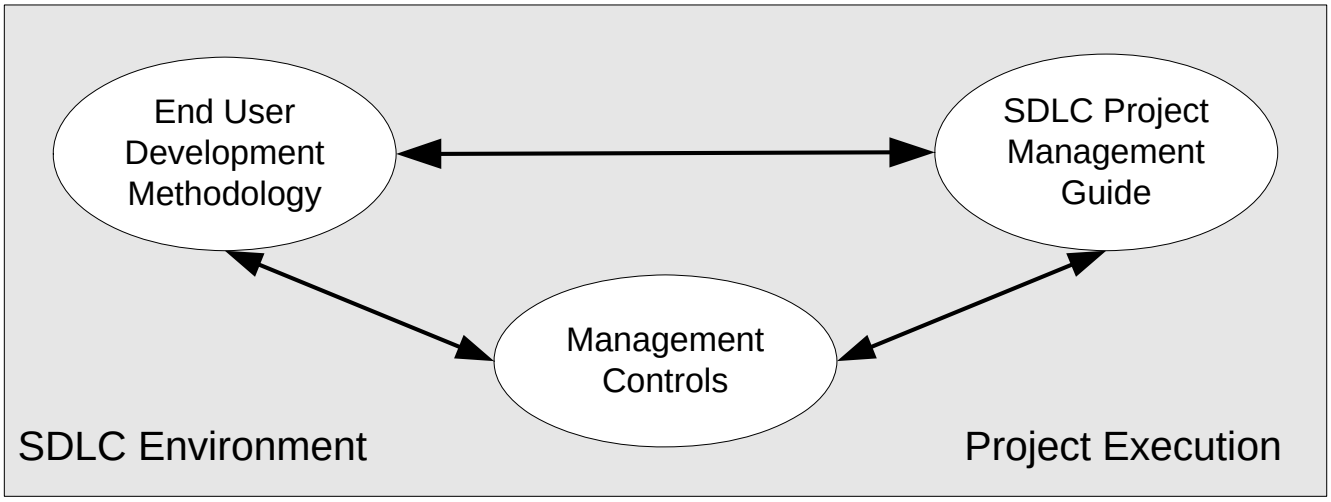
Integration and Test – User and system testing is conducted in this phase. The integration and test phase serves as an interim step before deploying a system to ensure minimal disruption during the deployment. The integration and test phase also serves as a means to collect feedback and make suggestions for refinements or adjustments prior to full deployment. For replacement systems or significant upgrades, data conversion may be required and will be conducted according to the implementation plan and verified through system/data testing. The PERS environment requires

transition of new or replacement systems to be done with minimal disruption to daily PERS activity. Parallel operations and advance training is critical to ensure continuity of services. Testing should be conducted both for user acceptance and system specification.

Implementation – The implementation phase includes all necessary activity to procure, receive, configure, and install the new or revised system. For non-technical solutions, implementation may be limited to a new support process requiring a change in the business process. Training is conducted during this phase according to the training plan, which would have been developed in one or more of the previous phases. A "transition" or "cut-over" plan, including any necessary data conversion, will also be required to ensure a smooth transition to the new system without interrupting services. The impact of running old and new systems simultaneously should also be analyzed to determine if there would be excessive burden in operating expenses or personnel support. The key output of this phase is a successful transition to the new system with uninterrupted service.

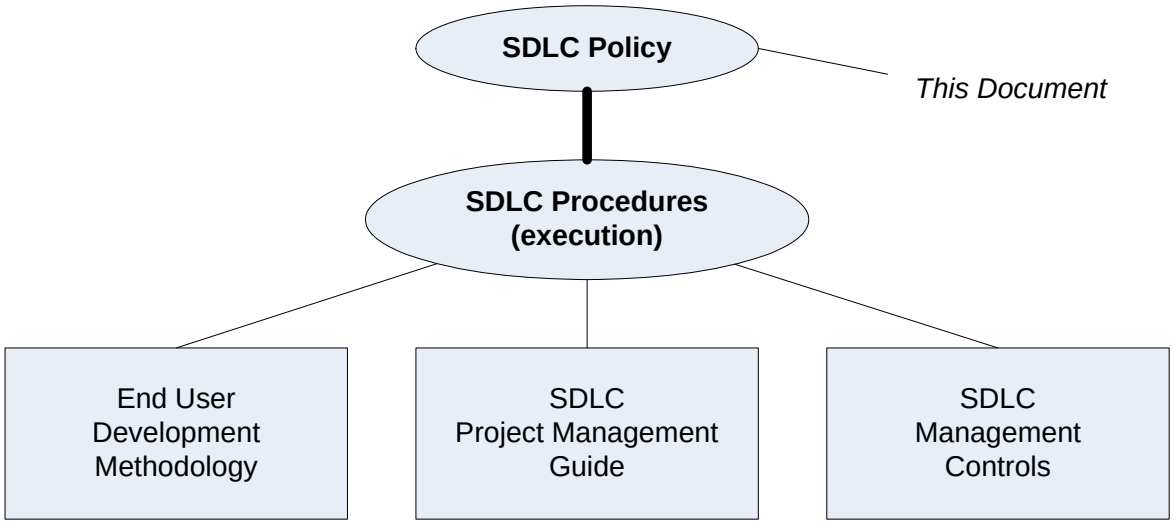
1.6. Relationships

End user development methodologies, project management guidance, and management controls are methods of addressing specific areas of SDLC implementation.



SDLC procedures provide more detailed guidance on implementing SDLC and specify preferred formats, template structures, or report content. Each of the three areas of the SDLC environment may also have detailed procedures developed to fit the organizations or groups using SDLC and will typically change or evolve with execution. An end user development methodology outlines details of PERS' approach to implementing SDLC consistent with industry best practices, which also evolves with technology and process change. Project management is a structured approach to managing the

use of methodologies to conduct projects or provide a support service. Management controls contain elements from both the end user development process as well as programmatic practices and ensure close coordination between the two. The relationship of this policy to specific procedures, methodologies, and guides is shown below.



The SDLC policy is supported by SDLC procedures established within the organizational area implementing SDLC or across the PERS community. The SDLC procedures are maintained in separate documentation and provide detailed guidance on the implementation of the SDLC process. Project management and end user development methodologies are more fluid in nature and are continually updated to reflect and take advantage of the best practices of industry.

Section 2 — SDLC Policy

The SDLC process outlined in this document provides a standardized method to help manage a wide range of activity to conduct projects or automate PERS' activities with information technology. This process can be used on any selected project whether or not technology is used to automate the activity. The SDLC policy requires execution of projects within the framework of the eight identified phases and tracking with a project notebook to reflect the SDLC status of projects at any given time. The practice of conducting structured systems development, by developing customer-based processes and solutions, promotes a smarter and less expensive way of doing business in PERS.

2.1 SDLC Key Terms

There are many different terms and expressions associated with systems development and project management. To provide a common basis for defining the SDLC phases and activities, some selected terms and definitions are provided to help develop the context for SDLC policy in the PERS environment.

Program/Project Manager

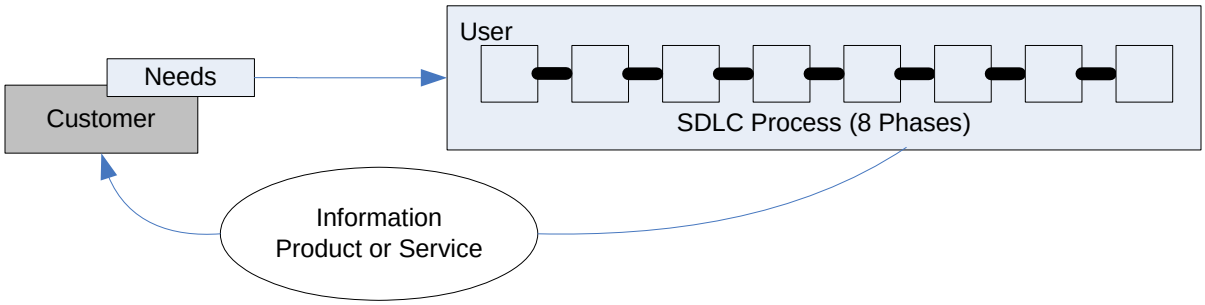
Program manager and project manager are two terms sometimes used interchangeably but actually differ in their roles. The program manager has overall responsibility for all projects within the program and the project manager is responsible for executing the task elements of the project. Designation of the program manager should be based on whether the project serves a dedicated group or serves an infrastructure role to support a wide user base where no particular group has ownership of the service or system. In the case of a dedicated type of project or system, the program manager should possess some background from the user perspective to ensure the user processes and requirements are fully represented in the project activity. For infrastructure type projects, the program manager can be selected from a wider experience base but should have experience in the technical area of the project and sound program management skills. The project manager will be selected for the specific skills necessary to execute the task elements of the project or simply possess the skills and experience to manage the work necessary to complete the project.

Program Manager – The program manager is responsible for managing programmatic activities such as personnel resources, budget, schedule, user involvement, and overall status of the program. For example, the program manager would be responsible to ensure items like a Statement of Work (SOW) and user requirements are included in a procurement package but the project manager would ensure the technical information is written. The program manager also ensures the business process dialog occurs between the users and system designers and coordinates any activity with other programs that have an impact or relation to program activity.

Project Manager – The project manager is responsible for development and execution of the task elements of the project. The project manager oversees day-to-day execution of the project and directs personnel in the accomplishment of tasks. The project manager is responsible for knowing what part of the systems development process the project is in at any time.

Customer/User

The second critical area for common understanding is the difference between customer and user and the relationship they have in the systems development process. The following diagram shows how the customer needs should be the basis for developing user requirements as part of the SDLC process.



The user processes and output should be clearly defined prior to any automation or enhancement which helps to ensure the output (results) is what the customer is looking for and the automation is user-friendly and an improvement to the process.

Customer – The customer is the ultimate benefactor of the service or information product. The customer needs should provide the primary influence on how the user constructs their operating concept and business processes.

User – The user (individual or group) that conducts the support processes, or uses the automation technology or system, to produce a service or product for the customer. The user has the key role of identifying the business processes and the areas for possible automation (new system) or enhancement (existing system or support process).

It is important to note that the customer can also be a user such as a member having access to a hypothetical Web-based finance system to view the status of their pay transactions. The finance group would be the primary user of such an automated finance system that serves the member customer base. The member, in addition to getting a paycheck as a customer, would also be a user of the system to "view only" the latest transactions for that member. This is an example of how the customer and user can be the same individual at times; highlighting the need to ensure that customer needs become the basis for developing user requirements.

2.2 SDLC Process

The SDLC process represents a PERS-wide coordination effort to conduct projects in a structured manner that promotes appropriate analysis necessary for project execution. The SDLC process provides a framework of phases that help track uniquely created tasks for each project. Tasks differ from activities in that they produce specific outputs such as decisions or documents on which further decisions can be made. Each project is assigned a project manager to develop the unique project tasks within each SDLC phase. The project manager keeps a project notebook that tracks the SDLC status of each project.

2.3 SDLC phases with key deliverables

The eight SDLC phases serve as a programmatic guide to project activity and provide a flexible but consistent way to conduct projects to a depth matching the scope of the project. Each of the eight SDLC phases is listed below with the basic intent, key deliverables, description of recommended tasks, and summary points for effective management controls. The management controls are cross-referenced to specific control objectives identified by Control Objectives for Information and related Technology (CobiT).

Initiation

Description: Careful oversight is required to ensure projects support strategic business objectives and resources are effectively implemented into an organization's enterprise architecture. The initiation phase begins when an opportunity to add, improve, or correct a system is identified and formally requested through the presentation of a business case. The business case should, at a minimum, describe a proposal's purpose, identify expected benefits, and explain how the proposed system supports one of the organization's business strategies. The presentation of a business case provides a point for managers to reject a proposal before they allocate any resources. Management should closely evaluate the necessity of each requested functional requirement. A single software feature approved during the initiation phase can require several design documents and hundreds of lines of code. It can also increase testing, documentation, and support requirements. Therefore, the initial rejection of unnecessary features can significantly reduce the resources required to complete a project.

Deliverables

Business Case document.

Major tasks

Project description – Provide a description of the project and outline the project purpose and objectives. Provide a simple operating concept diagram as appropriate to describe the project and other related areas or systems. Describe any required BPI/BPR efforts that may be necessary to validate or change the business process prior to automation initiatives.

Management controls (with cross-references to CobiT control objectives): The initiation phase should provide a horizontal look at the full project life cycle at a high level, but address certain key areas to manage project efforts within the strategic planning process. A summary of relevant control objectives includes:

- Project relevance to the organizational strategic plan
  - PO1.4 – IT Strategic Plan
  - ME4.2 – Strategic Alignment
- Roles and responsibilities of key groups related to the project
  - PO4.6 – Roles and Responsibilities

System Concept Development

Description: System concept development begins when the business case has been formally approved and requires study and analysis that may lead to system development activities. The main output of this phase is the creation of the Project Proposal document. The project proposal should identify alternative solutions and detail as many informational, functional, and network requirements as possible. The feasibility support documentation should be compiled and submitted for senior management or Board study. The Project Proposal document should provide an overview of the proposed project and identify expected costs and benefits in terms of economic, technical, and operational feasibility. Along with the description of alternative solutions, the document should include a recommendation for approval or rejection. The document should be reviewed and signed off on by all affected parties. If approved, management should use the project proposal and support documentation to begin the planning phase.

Deliverables

Project Proposal document (Feasibility Study).

Major tasks

Alternatives analysis – Provides an analysis of alternative approaches, including non-computer-based solutions such as support processes or procedural solutions, which reasonably achieve project objectives.

Cost/benefit analysis – Provides a summary analysis of the benefits of the project compared to the cost. May include return on investment analysis identifying the "break-even" point.

Management controls (with cross-references to CobiT control objectives): The system concept development phase should focus its efforts on providing the most accurate information to management so that an appropriate decision can be made on whether or not to proceed with a project. A summary of relevant control objectives includes:

- Personnel Resources
  - PO4.12 – IT staffing
  - PO4.13 – Key IT personnel
  - PO4.14 – Contracted staff policies and procedures
  - PO7.3 – Staffing of roles
  - DS3.4 – IT resources availability

- Project Proposal/Feasibility
  - AI1.3 – Feasibility study and formulation of alternative courses of action
  - AI1.4 – Requirements and feasibility decision and approval

Planning

Description: Many of the plans essential to the success of the entire project are created in this phase; the created plans are then reviewed and updated throughout the remaining SDLC phases. In the planning phase, the concept is further developed to describe how the business will operate once the approved system is implemented and to assess how the system will impact employee and customer privacy. To ensure the products and/or services provide the required capability on time and within budget, project resources, activities, schedules, tools, and reviews are defined.

Deliverables

Acquisition Plan
Configuration Management Plan
Project Management Plan

Risk Management Plan
Quality Assurance Plan
Validation and Verification Plan
System Security Plan

Major tasks

Develop Project Management Plans – Provide a schedule (Gantt Chart) of the project phases and major milestones, potential risks and mitigation strategies, quality measurements, and quality verification of work performed.

Management controls (with cross-references to CobiT control objectives): The planning phase should set out the course to be followed to successfully execute all phases of the project. A summary of relevant control objectives includes:

- Acquisition
  - AI5 – Procure IT resources
- Configuration Management
  - DS9 – Manage the configuration
  - DS10.4 – Integration of change, configuration, and problem management
- Project Management
  - PO10.2 – Project Management framework
  - PO10.3 – Project Management approach
  - PO10.6 – Project phase initiation
  - PO10.10 – Project Quality Plan
  - PO10.12 – Project planning of assurance methods
- Risk assessment
  - PO9.1–PO9.6 – Assess and manage IT risks
  - PO10.9 – Project Risk Management
  - DS2.3 – Supplier Risk Management
- Communications processes
  - PO6.5 – Communication of IT objectives and direction
- Quality Management
  - PO8.1 – Quality Management system
  - PO8.2 – IT standards and quality practices
- Validation and Verification Plan
  - AI7.2 – Test Plan
- System security
  - PO4.8 – Responsibility for risk, security, and compliance

- DS5 – Ensure systems security
- DS11.6 – Security requirements for data management

Requirements Analysis

Description: As mentioned above, the requirements analysis phase can be broken down into two parts: user requirements and system/data requirements. The user requirements portion of this phase requires the project manager to conduct a number of tasks leading to the development of a user requirements document. The customer needs will impact how the user establishes the mission, objectives, and operating concept for the support or service provided to the customer. Developing process flows using the Business Process Modeling (BPM) methodology at both the project level and user level will facilitate selecting the key processes to automate, which constitutes the user requirements. The details of the user requirements will be refined throughout the project. The system/data requirements portion of this phase is primarily dependent on the completion of tasks from the user requirements. Functional development areas are identified that facilitate technical activity or support cutting across many operational areas. This phase requires systems engineering skills and technical support to combine the user requirements and processes in a way that reduces redundancies, merges repetitive actions, and incorporates technology concepts. A well-defined system process that meets the user requirements is the basis for system/data requirements. The technical requirements will be a sub set of the system/data requirements and incorporate specifications of existing equipment or systems. The system/data requirements should clearly describe what part of the system process should be automated or enhanced—this helps promote a solution that meets the needs of many areas. The development of system processes and system/data requirements is the basis for design. The establishment of test criteria and test planning begins in this phase as well. Areas where testing will take place and who is responsible for the testing is included in the Test and Evaluation Master Plan.

Deliverables

User Process Flow diagrams (using BPM methodology) See attachment at end of this policy.
System Process Flow diagrams (using BPM methodology) See attachment at end of this policy.
Software Requirements Specification (SRS)
Interface Control document
Test and Evaluation Master Plan

Major tasks

Functional requirements – The core business processes are usually divided into major groupings called Functional Operating Areas (FOA). The primary processes within the FOA groupings are candidates for automation and become the functional requirements.

User process flows – The user process flows are developed through process engineering and may require interviews with the user community. The customer needs should have a significant impact on the development of user processes and be fully addressed by the user process.

User requirements development – The detailed activity of the user's processes serve as a basis for user requirements development. The user requirements should accurately describe what part of the user processes are candidates for automation or a support process. The user requirements should also include any security requirements and be validated to determine the final set of requirements eligible for the development effort.

System process flows – Involves development of system processes that combine user requirements and processes. System process flows are usually developed within Functional Development Areas (FDA), which provide a grouping of the automated technical activity. The systems approach allows elimination of redundancies and repetitive activities while also incorporating technical concepts or use of automated steps.

System/data requirements development – Provides information about which system processes should be automated or enhanced (capabilities and features) and identifies the operating environment and technical specifications. System performance parameters and database requirements are examples of information contained in the system requirements. A combination of technical information such as system requirements, database requirements, technical requirements, interface controls, system security, and other information comprise the system specification.

Management controls (with cross-references to CobiT control objectives): The requirements analysis phase should examine a number of critical areas leading to fully defined user requirements. The system/data requirements, along with the user requirements, provide the critical information for design. Although planning and organizing continues throughout SDLC, the system/data requirements serve as the basis for determining a solution. A summary of relevant control objectives include:

- Compliance with external requirements
  - AI1.1 – Definition and maintenance of business functional and technical requirements
  - AI5.1 – Procurement control
  - AI5.3 – Supplier selection

- User participation
  - DS1.1 – Service level management framework
  - DS1.3 – Service level agreements
- Requirements validation
  - AI1.1 - Definition and maintenance of business functional and technical requirements
- Identify roles and responsibilities
  - PO4.6 – Roles and responsibilities
  - PO4.7 – Responsibility for IT quality assurance
  - PO4.8 – Responsibility for risks, security, and compliance
- Acquisition considerations
  - AI2.5 – Configuration and implementation of acquired application software
  - AI5.4 – Software acquisition
- Test and Evaluation Master Plan
  - AI7.2 – Test Plan

Design

Description: The design phase is based on reviewing the contents of the user and system/data requirements. It involves converting the informational, functional, and network requirements identified during previous phases into unified design specifications that developers use to script programs during the development phase. The resulting systems engineering effort could range from the selection and configuration of an existing software product to a comprehensive systems development initiative involving software engineering and customization. For non-technical solutions, the design may simply be a support process to be implemented over time. The design process, along with the results of technology evaluations, may prompt an adjustment in the business process to take advantage of the cost saving of COTS solutions. The most important aspects of this SDLC phase is to provide solution alternatives with trade-off analysis that fit within the PERS environment and meet the validated requirements. This phase also marks the development of the Conversion Plan, Implementation Plan, *Maintenance Manual*, *Operations Manual*, Training Plan, and *User Manual*.

Deliverables

Conversion Plan
System Design document
Implementation Plan
Maintenance Manual
Operations Manual
Training Plan
User Manual

Major tasks

System Design – A systems engineering process that uses the results of technology evaluation to develop a solution or support process. The system design is structured to produce capabilities when combined with procedures provides solutions for enhancing performance or a support process. The security and internal control requirements, tailored to the PERS environment, are also embedded in the design process and provide a combination of technology and procedures to achieve the desired security protection and sound internal control environment.

Documentation – The process of formally recording information about the configuration of the system and the procedures for using the system or support process to solve problems and enhance operations. Typical documents include user manuals, operations manuals, systems guides, interface control documents, and maintenance guides.

Management controls (with cross-references to CobiT control objectives)

The design phase consists primarily of structured processes to meet the approved system design. A summary of relevant control objectives includes:

- Feasibility studies
  - AI1.3 – Feasibility study and formulation of alternative courses of action
- Comparison and risk reports
  - AI1.2 – Risk analysis report
- Information architecture
  - PO2.1 – Enterprise Information Architecture Model
  - AI2.1 – High-level design
  - AI2.2 – Detailed design
- Relevance to strategic technology goals
  - ME4.2 – Strategic alignment
  - PO4.2 – IT Strategy Committee
- Implementation Plan
  - AI7.3 – Implementation Plan
  - AI7.5 – System and data conversion
- *Operations and Maintenance Manual*
  - DS13.1 – Operations procedures and instructions
  - DS13.2 – Job scheduling
  - DS13.3 – IT infrastructure monitoring
  - DS13.5 – Preventive maintenance for hardware
- *User Manual*
  - AI4.1 – Planning for operational solutions

- Training Plan
  - DS7.1 – Identification of education and training needs

Development

Description: The development phase involves converting design specifications into executable programs. Effective development standards include requirements that programmers and other project participants discuss design specifications before programming begins. The procedures help ensure programmers clearly understand program designs and functional requirements. The development phase contains activities for building the system, testing the system, and conducting functional qualification testing to ensure the system functional processes satisfy the functional process requirements in the FRD. This phase addresses creating the contingency plan to ensure that appropriate steps are taken to maintain continuous service of major IT services on key business functions and processes. An integration document is also created in this phase explaining how the software components, hardware components, or both are combined and the interaction between them. At the end of this phase, the system will be ready for the activities of the integration and test phase.

Deliverables

Contingency Plan
Software Development document
System (application) software
Test files/data
Integration document

Major tasks

System build – Provides a controlled environment to conduct procurement, receiving, staging, configuration, and other build-out activities on a smaller scale. The build-out process also allows validation of vendor equipment and testing within a controlled environment prior to deployment in a production mode.

Management controls (with cross-references to CobiT control objectives): The analysis and design/system build phase consists primarily of structured processes to meet the approved system design. A summary of relevant control objectives includes:

- Technology evaluations
  - AI3.1 – Technological Infrastructure Acquisition Plan

  - AI3.2 – Infrastructure resource protection and availability
  - AI3.3 – Infrastructure maintenance
  - AI5.6 – Acquisition of infrastructure facilities and related services
  - PO3 – Determine technological direction
- Development procedures
  - AI2.1 – High-level design
  - AI2.2 – Detailed design
  - AI2.3 – Application control and audit-ability
  - AI2.6 – Major upgrades to existing systems
  - AI2.7 – Development of application software
  - AI2.10 – Application software maintenance
- Documentation procedures
  - AI2.2 – Detailed design
  - AI6.5 – Change closure and documentation
- System software security
  - AI3.1 – Technological Infrastructure Acquisition Plan
  - AI3.2 – Infrastructure resource protection and availability
  - AI3.3 – Infrastructure maintenance
- Ensure continuous IT service
  - DS4.1 – Continuity framework
  - DS4.2 – Continuity plans
  - DS4.3 – Critical IT resources
  - DS4.4 – Maintenance of the IT continuity plan
  - DS4.5 – Testing of the IT continuity plan
  - DS4.6 – IT continuity plan testing
  - DS4.8 – IT services recovery and resumption
  - DS4.9 – Offsite backup storage
  - DS11.5 – Backup and restoration

Integration and Test

Description: The objective of this phase is to prove that the developed system satisfies the requirements defined in the Software Requirements Specification (SRS). Several types of tests will be conducted in this phase. First, subsystem integration tests will be executed to prove that the program components integrate properly into the subsystems and that the subsystems integrate properly into an application. Next, system tests will be conducted to ensure the developed system meets all technical requirements, including performance requirements. Next, security tests will be conducted to validate that the access and data security requirements are met. Finally, users participate in acceptance testing to confirm that the developed system meets all user requirements as stated in the FRD. Acceptance

testing will be done in a simulated "real" user environment with the users using simulated or real target platforms and infrastructures. Testing in this phase includes validating the usability of the system or support process through reports such as test analysis, security evaluations, and system accreditation. System accreditation is the formal process for determining if the system meets user expectations (user acceptance) as outlined by the user requirements.

The testing phase requires organizations to complete various tests to ensure the accuracy of programmed code, the inclusion of expected functionality, and the interoperability of applications and other network components. Thorough testing is critical to ensuring systems meet organizational and end-user requirements. If organizations use effective project management techniques, test plans will be completed while developing applications, prior to entering the testing phase. Test plans created during initial project phases enhance an organization's ability to create detailed tests. The use of detailed test plans significantly increases the likelihood that testers will identify weaknesses before products are implemented.

Deliverables

Test Analysis report/approval determination
Test Problem report
User Acceptance or System Accreditation document/report

Major tasks

Test Analysis report/approval determination – The report documents each test – unit/module, subsystem integration, system, user acceptance, and security. Attached to the test analysis report is the approval determination as a final result of the test reviews and testing levels. These briefly summarize the perceived readiness for migration of the software.

System Accreditation – Final acceptance of the system is formalized through the system accreditation process, which provides acknowledgment that the user and system/data requirements have been met. The results of security evaluation and analysis are included in this area and must be acceptable to complete system accreditation.

Management controls (with cross-references to CobiT control objectives)

The integration and test phase requires monitoring of all activities ensuring that the testing process is complete and thorough. Summarized control objectives for this phase include:

- Installation procedures
  - AI3.1 – Technological Infrastructure Acquisition Plan

  - AI3.2 – Infrastructure resource protection and availability
  - AI3.3 – Infrastructure maintenance
- Change management procedures
  - AI6.1 – Change standards and procedures
  - AI6.2 – Impact assessment, prioritization, and authorization
  - AI6.5 – Change closure and documentation
- Validation, verification, and testing strategy/procedures
  - AI7.4 – Test environment
  - AI7.6 – Testing of changes
  - AI7.7 – Final acceptance test
  - AI7.8 – Promotion to production
- Process monitoring
  - AI7.11 – Recording and tracking of changes

Implementation

Description: The implementation phase involves installing approved applications into production environments. The phase is initiated after the system has been tested and accepted by the users. Primary tasks include announcing the implementation schedule, training end users, and installing the product. Additionally, PERS staff will input and verify data, configure and test system and security parameters, and conduct post-implementation reviews. Implementation schedules will be circulated to all affected parties and will notify users of any implementation responsibilities. A determination will need to be made by management whether to run the new system in parallel with the old system until the accuracy and reliability of the system is validated.

Deliverables

Delivered system
Post-implementation review

Major Tasks

Implementation – Development and processing of all required documents to procure equipment or services to deploy the system or support process. Additional activities may include staging, integration, configuration, and installation.

Training – Training should be conducted according to the training plan and provide a “self-sustaining” capability to PERS. The training curriculum should support users as well as instructors to ensure new users are afforded an immediate and continuous training opportunity.

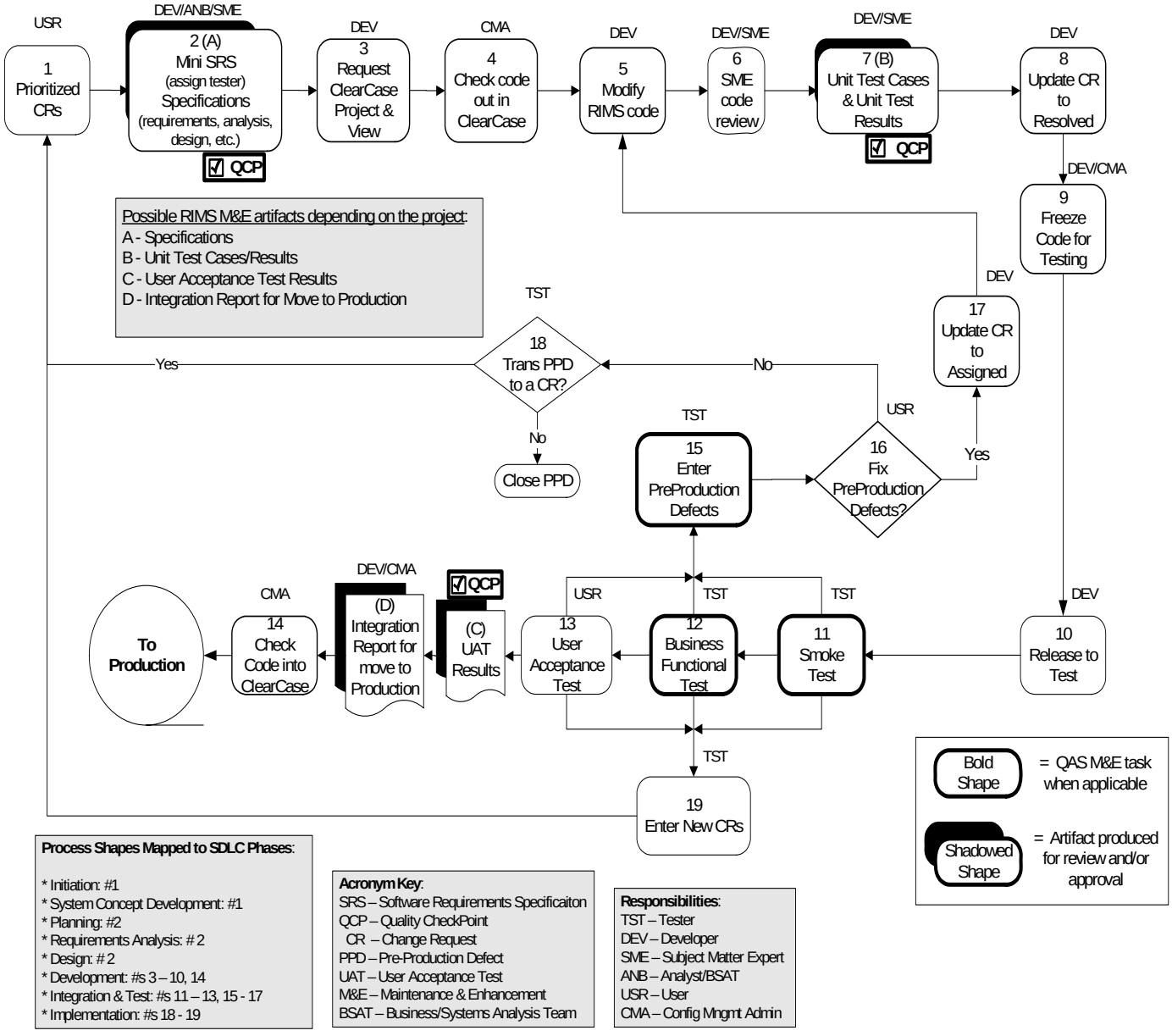
Post-implementation review – A post-implementation review shall be conducted to ensure the system functions as planned and expected, to verify that the system cost is within the estimated amount, and to verify that the intended benefits are derived as projected.

Management controls (with cross-references to CobiT control objectives)

The implementation phase requires diligent monitoring of all activities contributing to system deployment. Summarized control objectives for the deployment process to be continuously reviewed for applicability to the project include:

- Executing the Training Plan
  - AI4.3 – Knowledge transfer to end users
  - AI4.4 – Knowledge transfer to operations and support staff
  - AI7.1 – Training
  - DS7.1 – Identification of education and training needs
- Post-implementation review session
  - AI7.12 – Post-implementation review

RIMS Modifications/M&E Process Flow

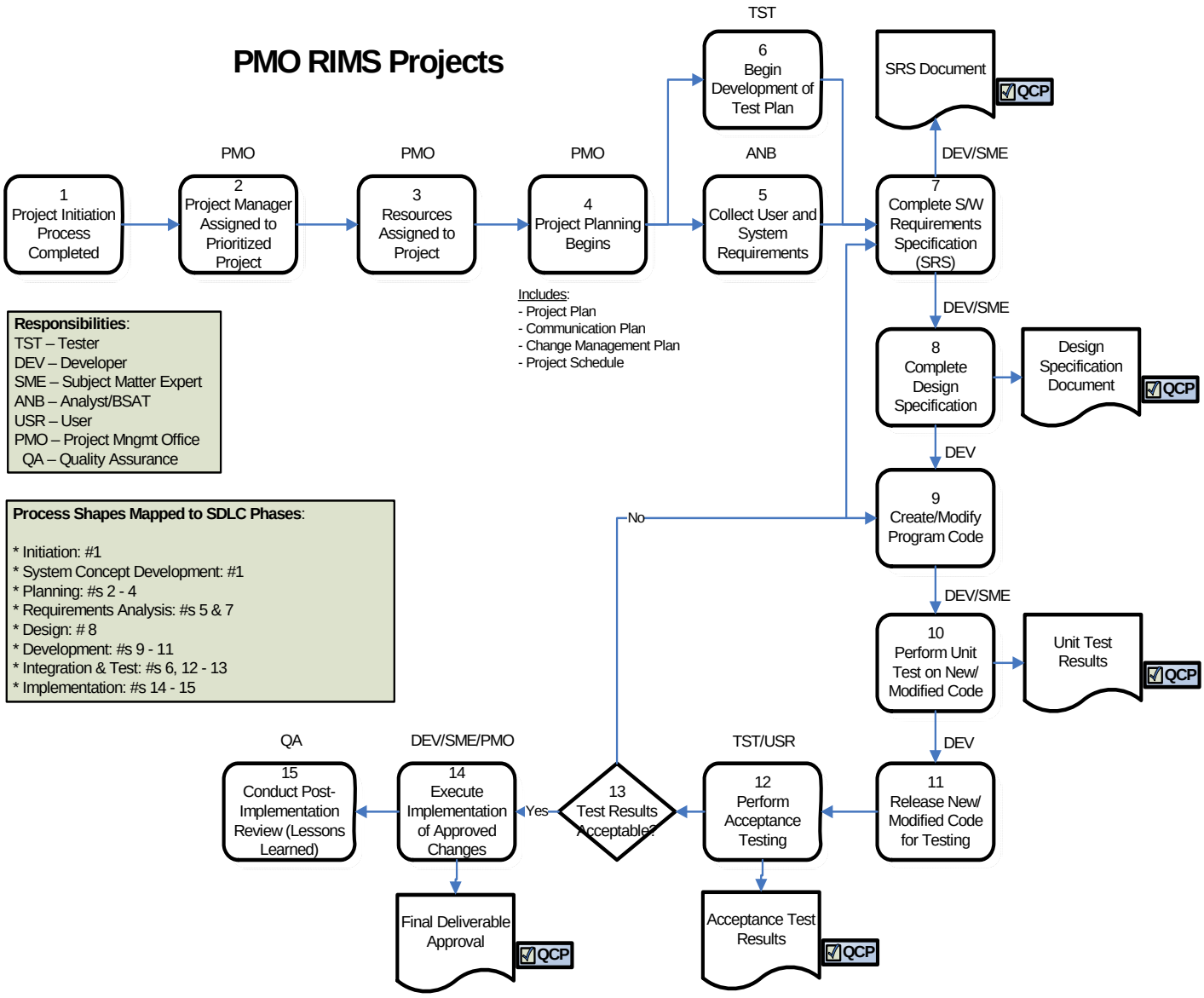


Process Shapes Mapped to SDLC Phases:

* Initiation: #1
* System Concept Development: #1
* Planning: #2
* Requirements Analysis: # 2
* Design: # 2
* Development: #s 3 – 10, 14
* Integration & Test: #s 11 – 13, 15 - 17
* Implementation: #s 18 - 19

Acronym Key:
SRS – Software Requirements Specificaiton
QCP – Quality CheckPoint
CR – Change Request
PPD – Pre-Production Defect
UAT – User Acceptance Test
M&E – Maintenance & Enhancement
BSAT – Business/Systems Analysis Team

Responsibilities:
TST – Tester
DEV – Developer
SME – Subject Matter Expert
ANB – Analyst/BSAT
USR – User
CMA – Config Mngmt Admin


PMO RIMS Projects
TST
6
Begin Development of Test Plan
SRS Document
QCP
PMO
PMO
PMO
ANB
DEV/SME
1
Project Initiation Process Completed
2
Project Manager Assigned to Prioritized Project
3
Resources Assigned to Project
4
Project Planning Begins
5
Collect User and System Requirements
7
Complete S/W Requirements Specification (SRS)
Includes:
- Project Plan
- Communication Plan
- Change Management Plan
- Project Schedule
DEV/SME
8
Complete Design Specification
Design Specification Document
QCP
Responsibilities:
TST – Tester
DEV – Developer
SME – Subject Matter Expert
ANB – Analyst/BSAT
USR – User
PMO – Project Mngmt Office
QA – Quality Assurance
DEV
9
Create/Modify Program Code
No
Process Shapes Mapped to SDLC Phases:
* Initiation: #1
* System Concept Development: #1
* Planning: #s 2 - 4
* Requirements Analysis: #s 5 & 7
* Design: # 8
* Development: #s 9 - 11
* Integration & Test: #s 6, 12 - 13
* Implementation: #s 14 - 15
DEV/SME
10
Perform Unit Test on New/ Modified Code
Unit Test Results
QCP
DEV
QA
DEV/SME/PMO
TST/USR
15
Conduct Post-Implementation Review (Lessons Learned)
14
Execute Implementation of Approved Changes
Yes
13
Test Results Acceptable?
12
Perform Acceptance Testing
11
Release New/ Modified Code for Testing
Final Deliverable Approval
QCP
Acceptance Test Results
QCP

Oregon Public Employees Retirement System
Business Process Modeling Methodology

April 10, 2006

Identification

Client:	Oregon Public Employees Retirement System
Project:	RIMS Conversion Project
System:	Clarety - Java Edition
Stage:	All Stages
Track Phase:	
Workflow Code:	BM
Artifact Type Name:	Business Process Modeling Methodology
Artifact Instance Description	
Major Release ID (Version):	
Minor Release ID (Draft):	
Artifact ID:	Business Process Modeling Methodology
Review Type:	*Formal*
Preparing Organization/Company:	PERS
Author's Name, Title/Position/Role:	Helen Perkins, Project Manager
Contract Number:	
Release/Approval Date:	
Period Covered Dates:	
Created Date:	March 6, 2006
Last Modified Date:	
File Name:	BPM Methodology.doc

Revision History

Change Request	Version Number	Date	Brief Description	Changed By
	1.0		Initial draft	

Table of Contents

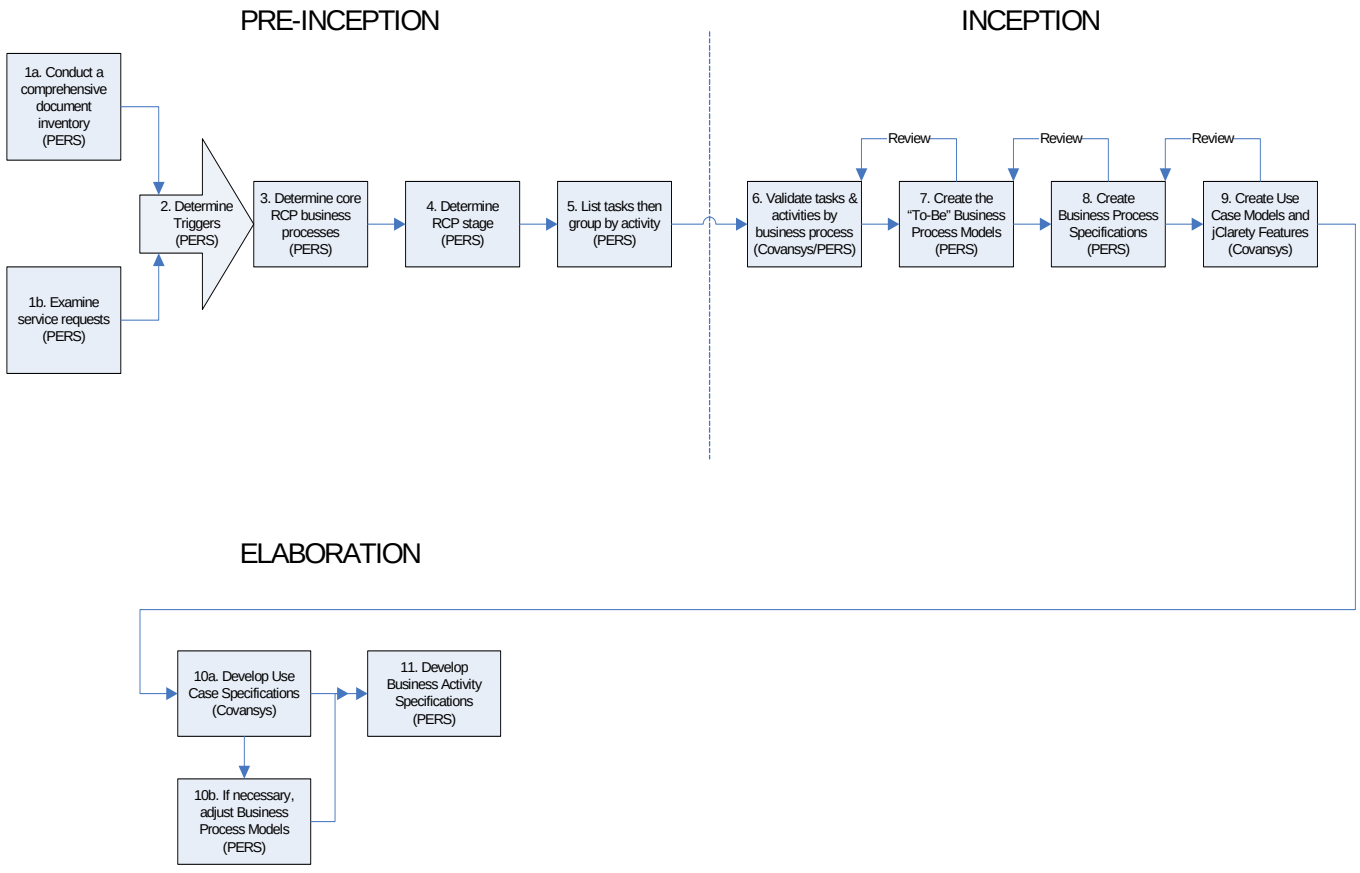
Introduction

The Business Process Modeling Methodology includes the steps that will guide the Business Process Owners (BPOs) and the Business System Analyst Team (BSAT) through the modeling effort resulting in Business Process Models and the Business Process Specification that will supply the necessary information to Covansys, enabling them to create the Use Case Models and jClarety features.

Another goal of the methodology is that while developing these models the level of detail is appropriate so that workflows can be easily developed.

Modeling Steps

The overall concept of this methodology is to keep it ***simple*** and not produce a document that is extensive which sits on the shelf gathering dust. However, enough detail is included in order to successfully develop the Business Process Models.



Pre-Inception Steps

1. **PERS:** Conduct a comprehensive document inventory (forms, letters, system reports) and examine service requests (phone calls, walk-ins, emails) and date-driven triggers that require action.
2. **PERS:** Determine triggers from step 1.
3. **PERS:** Determine core RCP business processes (RCP is the focus but other processes may be identified).
4. **PERS:** Determine the RCP Stage that the business process is in.
5. **PERS:** List all of the tasks involved in the business process. List the WHAT not the HOW. All of the tasks need to be documented so that we don't forget any of the "To-Be" tasks that may need to occur. Group tasks into activities:
    - Refine groupings for breaks in timing
    - Refine groupings for separation of duty
    - Refine groupings for other factors such as expected skill levels
    - Identify manual (business activity specification) vs. automated (use cases)
    - Identify systems that support automated tasks

Inception Steps

6. **Covansys/PERS:** Facilitated session to validate tasks and activities
7. **PERS:** Create the "To-Be" Business Process Models.
    - Includes role and control of responsibility
8. **PERS:** Create Business Process Specifications.
    - Includes activity, role, and tasks.
9. **Covansys:** Create Use Case Models and jClarety Features which link to PERS' Stakeholder Needs.

Elaboration Steps

10. **Covansys:** Develop Use Case Specifications and **PERS**: Adjust Business Process Models, if necessary.
11. **PERS:** Develop Business Activity Specifications.

The document inventory/service request examination (step 1) is the basis for defining all current business processes. All of the identified triggers should correspond to a business process or activity. Once all of the business processes have been identified, the RCP Stakeholder Needs can then be matched to a business process or activity.

General Modeling Guidelines

One of the most important concepts of process modeling is to know how much detail is needed. It usually does not add any value to model down to the lowest task levels just because the detail exists. Besides being time consuming, over analyzing a process often clouds understanding.

The following information provides guidelines for how much detail to include in the Business Process Model:

- Don't try and create the perfect model
- Understand the difference between what is understood (known) about the process and needs to be modeled to help others understand
- Keep it simple and on one page
- Push work up, not down
- An activity should be single user—group of tasks performed contiguously by the same role, using a single system.

Notation and Tool

A common set of symbols (notation) will be used to develop the Business Process Models. Everyone involved in the business modeling process needs to understand and use the symbols in the same way.

The common set of symbols that PERS will use is from the Unified Modeling Language™ (UML). The UML notation allows the modeler to specify, visualize, and construct the business process models.

The tool that the BSATs will use to construct the models using UML notation is IBM RequisitePro and IBM Rational Software Modeler. IBM RequisitePro is a powerful, easy-to-use requirements management tool that helps to manage requirements comprehensively. This tool offers the power of a database and Microsoft Word allowing PERS to easily organize and prioritize requirements, trace relationships between them, and track changes that affect the requirements and business process models.

IBM Rational Software Modeler is the UML 2.0-based visual modeling and design tool that will enable PERS to clearly document and communicate the business process models.

The ultimate goal using these tools is to develop Business Process Models that are understandable to all involved (BPOs, BSAT, Covansys).

Business Process Models

A business process is a collection of activities designed to produce a specific output in response to a trigger. A business process is thus a specific ordering of work activities across time and place, with a beginning, an end, and clearly defined inputs and outputs (may cross organizational boundaries). The model depicts the roles performing the activities, as well as the communication between the activities.

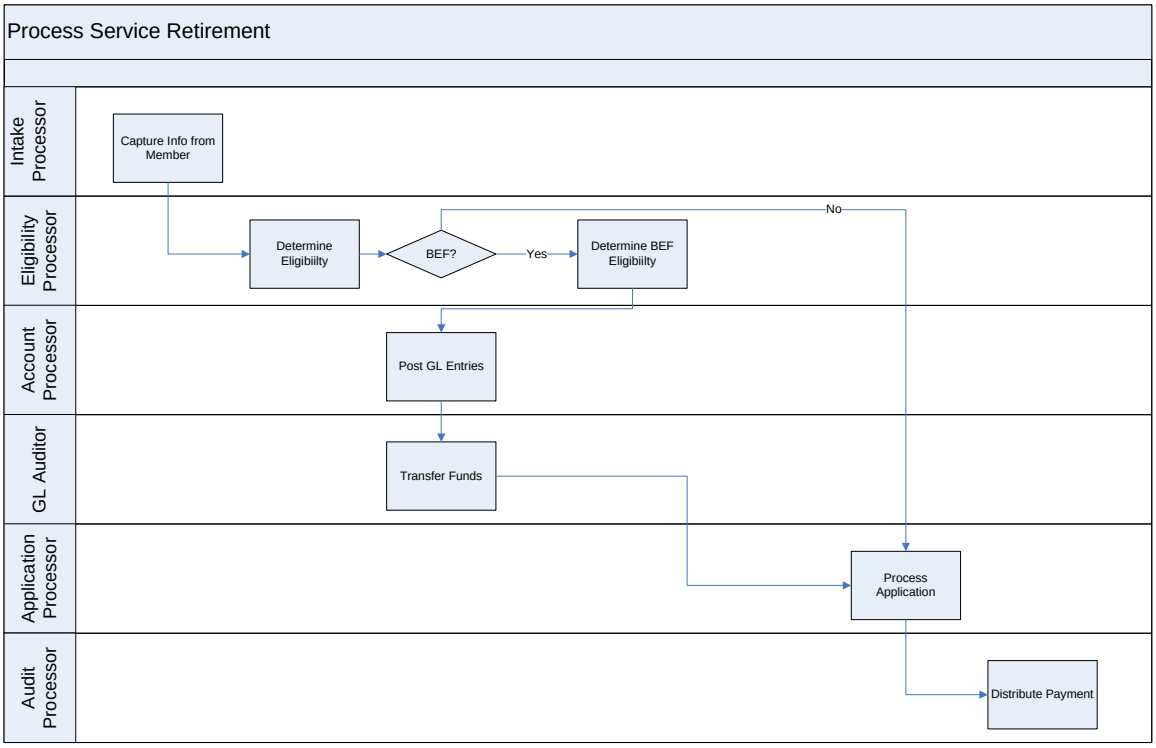
Guidelines:

1. An Activity is single user—group of tasks performed contiguously by the same role, using a single system (jClarety, RStars, etc.).
2. Swim lanes are at the role level (not at the department level). The information will be used to define jClarety and Workflow security groups.
3. The model heading is flush to the left. The model starts with a trigger in the form of a source as identified in the document inventory/service request examination.
4. Models end with a deliverable or trigger that is the input to a sink or contextual object from a higher-level model.
5. There are no more than 3-8 activities within the model. This helps to reduce complexity and makes the diagrams easier to understand. This is not a "task" diagram and should not be modeled to that level of detail.
6. Keep activities within the model uniform in size and shape.
7. Swim lanes represent roles that are internal or external to the business process under study:
    a. The source appears in the top swim lane.
    b. Internal swim lanes (roles internal to the business process) will be kept together in the central portion.
    c. System swim lanes are at the bottom and should only be defined if it is a "batch process."
8. Activities flow from the upper left to the lower right.
9. An activity will always have at least one input (trigger) and one output (deliverable).
10. When using a converging synchronization bar before an activity, it must indicate multiple deliverables are needed before an activity can begin.
11. Sources and Sinks will be used sparingly to refer to otherwise distracting or out-of-scope information.
12. Stores are used to represent systems that receive deliverables and either process, store, maintain, and/or forward those deliverables to another activity.
13. See Appendix A – Business Process Model Checklist for additional elements that should be included in the model.

Inputs:

1. Inventory of forms, letters, reports, service requests used in the business process
2. Stakeholder Needs
3. Business Rules
4. List of roles participating in the process
5. Current procedures
6. Definition of how RIMS currently supports the process
7. An understanding of how current production jClarety activities support the process
8. Define other standalone systems that support the process

Example Business Process Model



Model Components:

Activity – An identifiable piece of work, typically done in one sitting by a single role or multiple cooperating roles, which consists of one or more tasks. These are component steps in the business process, performed by a role. Activities show a role's involvement in a process and the deliverable from that involvement.

Control Flow – The hand-off of control from one activity to the next is the focus of the model. Each activity represents work performed by a role for a period of time. Once the role completes its responsibility, control of the process is passed to the next activity. The control flow represents the passing of this control, and may also contain two components. The first is a deliverable which represents the work product of the source activity used by the subsequent activity. The second component is a trigger that initiates the recipient activity. This trigger may be as simple as the completion of the source activity.

Decision Point – The flow of work through the business process may not be uniform each time it is initiated, but can vary based upon conditions. A decision point evaluates a specified condition during the process, and directs the flow of work based upon its evaluation. A decision point receives a single path and has multiple possible paths to subsequent activities based upon the number of alternatives for the condition.

Role – Roles (roles and systems) participate in the model as internal or external to the business process under study. Internal roles are roles that perform the activities within the scope of the model. External roles are outside the control of the business process, but interact with the process by providing resources, or receiving products and services. The external role's activities are typically at the beginning or end of the model.

Store – An aggregation of deliverables maintained by, and provided to, activities in a business process. A store may be the destination that holds their deliverables for subsequent use. A store

may also be the source that passes deliverables which have been previously stored. An originating store may also serve the purpose of a queue.

Source – An entry point to the model. A source is used to display a deliverable received by an activity, whose previous activity or origin is unknown.

Sink – An exit point from the model. A sink is used to display a deliverable produced by an activity that has an unknown subsequent activity or destination.

Synchronization Bar – A diagramming construct that allows the model to converge/diverge into another model. Synchronization bars are used to combine or split deliverables as they move from one level in the process hierarchy to another.

Appendix A – Business Process Model Checklist

The following should be submitted to Covansys:

Business Process Model	
Business Process Specification	
Mapped to Stakeholder Needs	
Samples of forms, letters, reports, service requests used in the business process	

Required Guidelines and Elements:

Modeling Guidelines:	
▪ Activities are single user, single setting groups of tasks	
▪ Swim lanes are at the role level	
▪ Start with a trigger and end with a deliverable or trigger	
▪ There should be no more that 3-8 activities	
▪ Interactions with external entities are defined.	
JClarety:	
▪ Describe JClarety usage (1 sentence maximum)	
▪ List of inputs and outputs	
▪ List of Business Rules used	
▪ List of "exception" paths through the activity:	
➔ Variances that rely on choices the user or the customer may make (e.g. a member applying for retirement may choose to waive his/her benefit). A separate model is provided for a variant when the exception path differs significantly from the main flow.	
➔ Scenarios that should result in a different outcome—e.g. a member applying for retirement may not be eligible.	
Workflow:	
▪ Milestones to show workflow progress	
▪ Metrics that should be reported	
▪ Desired use of workflow and imaging is identified	
▪ Audit controls have been identified, where applicable	
▪ Security needs have been captured, where applicable	
RIMS usage is modeled where used	
All terms are used consistently and are defined in a glossary.	

Appendix B – Business Process Specification Template

Template

Appendix C – Glossary of Terms and Acronyms

TERM	DEFINITION
Activity	An identifiable piece of work, typically done in one sitting by a single role or multiple cooperating roles, which consists of one or more tasks. These are component steps in the process, performed by a role. Activities show a role's involvement in a process and the deliverable from that involvement.
Artifact	A piece of information that is produced, modified, or used by a process; defines an area of responsibility; and is subject to version control. An artifact can be a model, a model element, or a document.
Business Activity Specification (BAS)	PERS Use Case for manual activities.
Business Area	A collection of functions that share a common objective; component of a Business Domain. Example: Provide Membership Services [Domain] Provide ER Services [Area].
Business Domain	A cohesive portion of the business used as the scope for analysis includes one or more Business Areas. Examples: Provide Membership Services [Domain]
Business Function	A collection of interrelated processes; component of a Business Area. Example: Provide ER Services [Area] Provide New ER Coverage [Function].
Business Process	A business process is a collection of activities designed to produce a specific output in response to a trigger. A business process is thus a specific ordering of work activities across time and place, with a beginning, an end, and clearly defined inputs and outputs.
Business Process Model	Represents the business process in terms of its component activities and the flow of work among the activities (may cross organizational boundaries). The model depicts the roles performing the activities, as well as the communication between the activities.
Business Process Specification	The documentation that is provided with the Business Process Model.
BPO	Business Process Owner
BSAT	Business System Anaylst Team
Control Flow	The hand-off of control from one activity to the next is the focus of the model. Each activity represents work performed by a role for a period of time. Once the role completes its responsibility, control of the process is passed to the next activity. The control flow represents the passing of this control, and may also contain two components. The first is a deliverable which represents the work product of the source activity used by the subsequent activity. The second component is a trigger that initiates the recipient activity. This trigger may be as simple as the completion of the source activity.
Customer	Recipient or beneficiary of the result produced by the business process. Also known as constituent.

	process. Also known as constituent.
Decision Point	The flow of work through the business process may not be uniform each time it is initiated, but can vary based upon conditions. A decision point evaluates a specified condition during the process, and directs the flow of work based upon its evaluation. A decision point receives a single path and has multiple possible paths to subsequent activities based upon the number of alternatives for the condition.
Deliverable	A work product that is ultimately provided to customers outside of the boundaries of the business domain, and are therefore the reasons a business domain exists.
EBP	Essential Business Process.
IBM Rational Software Modeler	The UML 2.0-based visual modeling and design tool that will enable PERS to clearly document and communicate the business process models.
IBM RequisitePro	Powerful, easy-to-use requirements management tool that helps to manage requirements comprehensively.
Model	A semantically closed abstraction of a system. A complete description of a system from a particular perspective. Complete: meaning you don't need any additional information to understand the system from that perspective. A set of model elements. Two models cannot overlap.
Organizational Diagram	A hierarchical model used to relate defined groups of people or individuals. These business (non-system) roles include organizations, markets, roles, and people. An organizational diagram is typically used to define the vertical reporting structure of an organization. An organizational diagram typically places a broad market or organization at the highest level (or root) in the hierarchy. More specific roles within the general one are positioned below it. The lower in the hierarchy the role appears, the more detailed the role.
Process	See Business Process
Requirement	Describes a condition or capability that a system must provide; it is either derived directly from stakeholder needs, or stated in a contract, standard, specification, or other formally imposed document.
Role	Roles (roles and systems) participate in the model as internal or external to the business process under study. Internal roles are roles that perform the activities within the scope of the model. External roles are outside the control of the business process, but interact with the process by providing resources, or receiving products and services. The external role's activities are typically at the beginning or end of the model.
RUP	The **Rational Unified Process** (RUP) is an iterative software development process created by the Rational Software Corporation, now a division of IBM. RUP is not a single concrete prescriptive process, but rather an adaptable process framework. As such, RUP describes how to develop software effectively using proven techniques. While RUP encompasses a large number of different activities, it is also intended to be tailored, in the sense of selecting the development processes appropriate to a particular software project or development organization. RUP is recognized as particularly applicable to larger software development teams working on large

	projects.
Sink	An exit point from the model. A sink is used to display a deliverable produced by an activity that has an unknown subsequent activity or destination.
Source	An entry point to the model. A source is used to display a deliverable received by an activity, whose previous activity or origin is unknown.
Stakeholder	Individuals who are involved in or may be affected by RCP activities.
Stakeholder Need	RCP functionality that stakeholders have identified as essential, necessary, and/or desirable.
Store	An aggregation of deliverables maintained by, and provided to, activities in a business process. A store may be the destination that holds their deliverables for subsequent use. A store may also be the source that passes deliverables which have been previously stored. An originating store may also serve the purpose of a queue.
Swim lane	A partition on the model organizing the responsibilities for actions.
Synchronization Bar	A diagramming construct that allows the model to converge/diverge into another model. Synchronization bars are used to combine or split deliverables as they move from one level in the process hierarchy to another.
Task	A defined action that is directly associated with an activity. A step performed by or considered by a role while completing an activity. An activity is comprised of one or more tasks. Automation should be seen at this level.
Traceability	A relationship between two requirements that implies the source, derivation, or dependencies between the requirements using the "trace to" and "trace from" features.
Trigger	Initiates the business process or activity. Triggers can be a form, letter, report, service request, date-driven action, etc.
UML	Unified Modeling Language is an industry-standard language. The UML notation allows the modeler to specify, visualize, and construct the artifacts of software systems, including business models.
Use case	A description of system behavior, in terms of sequences of actions. A use case should yield an observable result of value to an actor. A use case contains flows related to producing the "observable result of value", including alternate and exception flows.