SECTION 4 – FULL-SCALE DEVELOPMENT LIFE CYCLE METHODOLOGY

Segments of the Life Cycle

I - MANAGING THE SOFTWARE DEVELOPMENT PROJECT 2

II – APPLICATION NOTIFICATION 3

III - FEASIBILITY STUDY 4

IV – REQUIREMENTS DEFINITION AND CONCEPTUAL DESIGN 14

V - ARCHITECTURE 23

VI - REQUIREMENTS SPECIFICATION AND PROTOTYPE 29

VII - PROGRAM DEVELOPMENT 36

VIII - UNIT TESTING 39

IX - PHASE TESTING 41

X - USER TESTING 43

XI – INTEGRATION TESTING 45

XII - SYSTEM TESTING 50

XIII - IMPLEMENTATION 52

XIV PRODUCTION SYSTEM SUPPORT 60

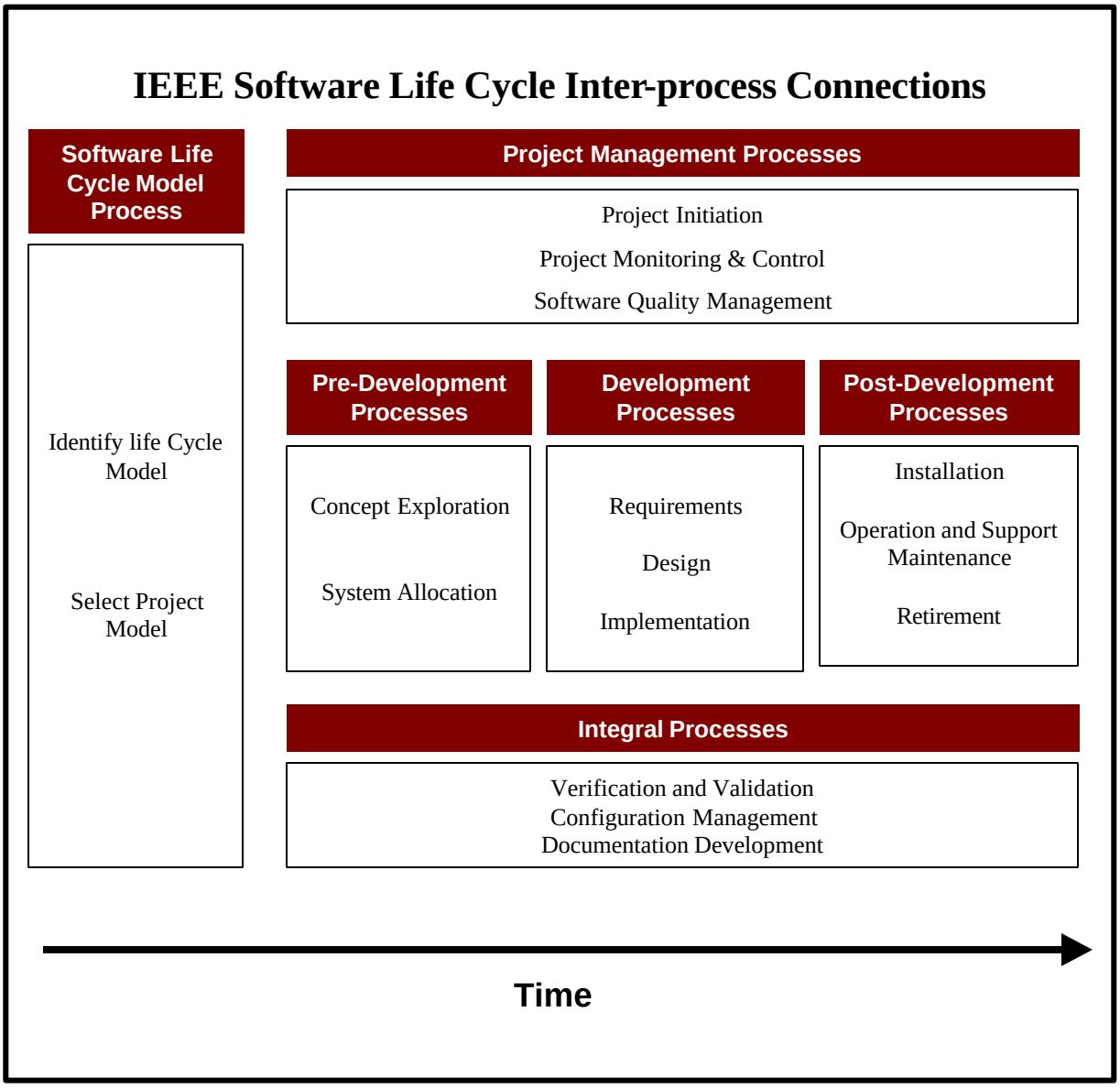
XV - PROCESS ASSESSMENT 63

XVI - CHANGE MANAGEMENT 65

I - MANAGING THE SOFTWARE DEVELOPMENT PROJECT

The software life cycle process integrates the following Project Management Processes: project planning, initiation, monitoring and control and quality management. The diagram below shows that project management spans the entire software development life cycle, from pre-development through post-development.

Software development projects must use a project management framework prescribed by DENR. A separate document (DENR Project Management Toolkit) describes this framework and is available on the DENR network.



II – APPLICATION NOTIFICATION

II.A Application Notification Purpose

The Notification is a formal document that identifies the desire to develop a new application or to significantly upgrade an existing application. This document gives a sense of the scope, complexity, and impact of the proposed application. It also notifies ITS about the proposed application where ITS may be able to provide assistance in project design and implementation.

II.B Application Notification Objectives

To identify proposed application

To identify requestor/sponsor individual and organization.

To notify ITS about the proposed application.

II.C Application Notification Output

1) Application Notification (see Reference II.C.1)

References

Reference II. C. 1 Proposed Application Notification (PAN)

1. To initiate an application development project, a PAN must be submitted to the Director of SDT and the ITS Director. This notification would include the following information:

    1. The name, address, phone number, organization and title of the sponsor.
    2. The name of the proposed application.
    3. If the proposal stems from a legislative mandate, which law?
    4. A name and general description of the business process(es) this project is expected to impact.
    5. The purpose of this application as it relates to the business process(es).
    6. The key internal and external customers this project is intended to benefit.
    7. If known, a high-level overview/proposal of the technology that might be used to develop the proposed application.
    8. If known, existing or proposed applications, databases, or technologies that this application might be expected to affect. These applications, databases, or technologies may reside inside or outside of the sponsor's immediate organization.

III - FEASIBILITY STUDY

III.A Feasibility Study Purpose

This is an initial study intended to determine the initial scope of the project and to provide information about the costs, benefits, and risks associated with the proposed application and the related development project. It is also intended to identify any benefits that may occur regarding staff productivity, customer satisfaction, or significant savings of supplies or materials. An overall assessment of these general factors should be undertaken to determine whether the project is worth pursuing.

III.B Feasibility Study Objectives

1) Identification and description of current business process(es) associated with the proposed application.

2) Based on a review of the relevant business process(es), a listing and description of the problems and/or opportunities the proposed application is intended to address.

3) Identification of reasonable alternative application solutions, listing of anticipated advantages and disadvantages for each of the solutions.

4) Selection of preferred solutions (no more than ~~two or~~ three) from the initial list of alternatives.

5) Estimation of the costs, benefits, and risks associated with this selection (these selections).

6) An initial statement of project scope

7) Final recommendation.

III.C Feasibility Study Output

Completing the steps outlined in Section III.D will result in a Feasibility Study Report that will provide the necessary background information on the project and a "go"/"no go" recommendation on the project. The Feasibility Study Report specifications are defined below:

1) Executive Summary - The executive summary would briefly describe the business process(es) and the related issues and opportunities to be addressed by the proposed application, a summary of the potential costs and benefits associated with the proposed application, an overview of the risks associated with the related development project, and a summary of the final recommendation.

2) Project Background - The project background should include:

A restatement of the initial notification

A description of the relevant business process(es) (See Reference III.C.1)

A description of issues and opportunities associated with the relevant business process.

A List of Common Processes that present reuse opportunities.

An explanation that describes how these issues and opportunities will be addressed by the proposed application.

3) Alternatives - The alternatives section should include:

A description of each alternative (See Reference III.D.4)

A listing of advantages and disadvantages for each alternative.

4) Recommendation - The recommendation section should include:

The selection of a preferred alternative with an analysis of advantages and disadvantages to justify the selection.

An initial Technical Capability Table comparing the preferred alternative(s) against current IRMC Standards.

A high level description of the development project associated with the proposed application.

An analysis of the benefits that are expected to come from the proposed application.

An analysis of the costs and risks associated with the selected solution and its related development project (See reference III.C.2)

6) Signatures - The signature section of this document will be signed by the ITS Director and the project sponsor. A statement should precede the signatures indicating that the ITS Director and project sponsor accept the Feasibility Study Report and its recommendation.

III.D Steps to Complete the Feasibility Study Report

Step	Description	Inputs	Outputs	Resources	Notes
1	Establish the Feasibility Study Team		Team Roster	Project Manager & Project Sponsor	See reference III.D.1
2	Prepare and obtain signatures approving the initial project scope	Application Notification	Approved Scope Statement	Project Manager & Project Sponsor and other client management where appropriate	See reference III.D.6

Step	Description	Inputs	Outputs	Resources	Notes
3	Develop an initial description of the relevant, existing business process	Application Notification, Any Existing Workflow or Business Process Documents	Initial Process Description and/or workflow diagram. For object-oriented dev't, actors list, use case list, high level use case model.	Team/process experts	See reference III.C.1 and reference III.D.2; See Appendix I for example workflow diagram
4	Identify Problems and Opportunities Associated with the Business Process	Process description and Associated Workflow Diagrams	Statement of Problems, Opportunities, List of Common Processes	Team/process experts	See reference III.D.3, III.D.7
5	Brainstorm Alternative Solutions and Related Advantages and Disadvantages	Process Description and/or Any Associated Workflow Diagrams, Statement of Problems, Opportunities, NC Statewide Technical Architecture	List of Alternatives with Respective Advantages and Disadvantages	Team/process experts /technology experts / IRM staff	See reference III.D.4
6	Select Preferred Alternative and Conduct Cost/Benefit/Risk Analysis	List of Alternatives with Respective Advantages and Disadvantages, Process Description and Associated Workflow Diagrams, Statement of Problems, Opportunities, IRMC Standard for Estimating Costs and Delivery Dates	Cost/Benefit/Risk Analysis and Recommendation,	Team	See reference III.C.2; III.D.4, III.D.5

Step	Description	Inputs	Outputs	Resources	Notes
7	Prepare Feasibility Study Report	Results from Previous Steps	Feasibility Study Report	Team	See Reference III.D.8
8	Obtain Approval and Commitment to Continue	Feasibility Study Report	Signed Feasibility Study Report	Team, Project Sponsor	

References

Reference III.C.1 Business process characteristics

Consider the following items when developing, validating, and refining your business process description:

1 A process name that fairly represents what the process does.
2. A list of inputs, outputs, and a description or diagram that lays out how inputs are transformed into outputs.
3. External "triggers" that initiate the process (e.g. a land developer initiated a permit application.)
4. Important data flows which represent data or physical materials (e.g. permit application) in motion, often represented as an arrow line diagram.
5. Data stores that describe data at rest (e.g. files, databases)
6. Terminations, the people or systems with which the process communicates or interacts.
7. Object oriented projects should also identify actors and high level use cases.

Reference III.C. 2 Possible Risk Factors

The following is a list of risk factors to consider when performing a risk assessment. Consider the probability of the risk occurring and the consequences of its occurrence. If the probability is medium - high with high negative consequences, determine actions that can be taken to mitigate the negative impact associated with the risk's occurrence.

New Technology	**Project Team**
Database Software	Level of Morale
Communications Software	Staff Availability
Programming Language	Commitment to Team
Tools or Aids	Technical Skills
Communications Network	Applications Knowledge
Mainframe	Staff Turnover
Testing Resource	Familiarity with each other
PC's or Desktops	Staff Conflicts
Information Center	Size of Team
Geographic Dispersion	
Reliability of Personnel	
Object Oriented Design	
State Service Broker	

Applications	**Organizational**
Project Size	Stability of IS Organization
Functional Complexity	Stability of User Organization
New (vs. Replacement)	Number of User Departments Involved
Quality of available information	Availability of Support Organization
Vulnerability to change	Extended Team Commitment
Stability of business need	Conflict Resolution Mechanism
Intensity of business need	Signoff/Approval Process
Organizational impact	Senior Management Commitment
Interface with other applications	Availability of Champion

Dependency on other projects	Continued Budget Availability
Conversion difficulty	Chargeback System
Tight timeframe	Project Standards used
End date dictate	Accountability for change
Organizational impact	
Interface with other applications	

User/Client	**Project Manager**
Level of Commitment	Problem Solving Skills
Attitude toward IS	Managerial Identity
Applications Knowledge	Influence Skills
IS Systems Knowledge	Achievement Drive
Change Propensity	Experience with User
Turnover of Key People	Experience with Application
User to Customer Relations	Experience with Organization
Readiness for Takeover	Experience with Technology
Design Participation	Experience with Project Team
Change Process	Planning Skills
Need for Organization Chart	Estimating Skills
	Possibility of Turnover

Complexity is a key variable when evaluating risk and should be considered from several perspectives:

Application size: number of modules, function points and entities accessed, number of transactions per period of time. Object oriented projects should also consider the number of actors and use cases.
Logic complexity: complexity of calculations, nesting of decisions, complexity of decision trees, string/text handling
Data complexity: simple read-only, complex query, simple update or complex update.
Database support: single physical dedicated database vs. many heterogeneous, physically disparate databases
Interface complexity; single character-based interface vs. portable, GUI interface.

The IRM has procured Risk Assessment tools that are available to the State agencies. Contact the IRM Quality Assurance staff to avail of this tool.

Reference III.D.1 Feasibility Team Membership

At a minimum, a core team should consist of the project sponsor and a project manager. It is also recommended that the team include staff who are expert in the relevant business process, IRM staff and technology experts who could assist in identifying alternative technology solutions and their respective advantages and disadvantages. These members may be temporary, participating when their expertise is required.

Contact the IRM Chief Technology Officer (CTO) or Deputy CTO to identify an IRM participant or IRM resource person for the project.

Reference III.D.2 Developing Initial Description of Business Process

Consider the following possible approaches as you develop a detailed description of the relevant process:

If process descriptions exist, present them to the Feasibility Study Team for

review. Conduct a detailed review of these materials keeping items in Reference III.C.1 in mind.

If a process expert exists who has a detailed knowledge of the entire process and has good analytical skills, have this person develop a preliminary workflow (assuming a workflow description is appropriate in characterizing the process) description including the items listed in Reference III.C.1 and present this description to the Feasibility Study Team for discussion.

If no such person exists, carefully select representatives who are well-versed in different general aspects of the process as well as a couple of staff who have a good general overview of the process. Then hold a series of facilitated discussions. The facilitator should be experienced specifically with process review and design.

This description is an initial description to assist in the development of a feasibility assessment. When detail user requirements are developed in the next section, this initial process description should be detailed further and validated. The steps for doing so will be discussed in the next section.

Reference III.D.3 Developing Statement of Problems and Opportunities

The same team that develops the initial process description, should identify the problems and opportunities. Many of the problems and opportunities may emerge during the initial discussion of the process. These should be recorded, possibly on a separate flip chart, and held for the subsequent discussion that is dedicated to identification of problems and opportunities.

Once the process description has been completed, the Feasibility Study Team and process experts should review the process description to identify problems and opportunities associated with the process. In particular, the team should:

- Look for steps in the process where such things as backlogs, bottlenecks, incidents, and complaints are known to occur.
- Also consider the process from the perspective of key customers. What kinds of services/outcomes would you want were you a customer? What steps in the current process either facilitate or hinder the realization of these services and outcomes? How?

Reference III.D.4 Identifying and Developing the List of Alternatives

When considering possible application solutions, first consider how the existing process can be changed to address problems and take advantage of opportunities inherent in the current process. Then engage in a facilitated brainstorming session(s) to identify alternative application solutions to enable the identified changes in the current process. This discussion would probably benefit from the inclusion of technology experts who are well-versed in state-of-the-art technology. The IT experts may be internal staff, IRM staff, ITS-C staff or external consultants.

1) Consider the following as you develop your list of alternative solutions:
2) the general architecture of the solution (e.g. n-tiered client/server); - and how it stacks up against the Statewide Technical Architecture (STA). Begin to use the Technical Capability Table as a checklist to assess the alternatives.
3) the kind of development tools required (e.g. DBMS, CASE, object oriented programming language) and that are available from the standard suite of

tools defined by the department,
4) networking requirements,
5) communications requirements,
6) processor requirements,
7) storage requirements,
8) security requirements (i.e., identification, authentication, authorization and access control requirements for users of the system; as well as administration and audit requirements)
9) The possible existence of reusable software within the department, or within other State agencies. Reusable software may be an entire application, a sub-function, or specific components such as shared services.

It is important to seek IRM involvement as early in the project cycle as possible since alignment with the Statewide Technical Architecture is a prerequisite to obtaining subsequent project certification from the IRMC. Key considerations when performing these activities under the Feasibility Study phase are:

- An understanding of the principles and policies defined in the Statewide Technical Architecture (STA) is a must, and provides the architectural direction when developing a list of alternatives. The Feasibility Study team must review the STA document prior to developing a list of alternatives.

- Consult the IRM (i.e. the participant or identified resource from IRM) as a source of information on:approach and technologies that other departments state agencies may be using or have used to address the same situation IRMC standardized solutions that state agencies must comply with Any lessons learned collected by the IRM related to the alternatives being considered by the department

As you identify alternative solutions, record any advantages and disadvantages that might emerge during the discussion. Once the alternatives list has been completed, review the advantages/disadvantages for each alternative supplementing the respective lists where appropriate.

Reference III.D.5 Cost/Benefit

The following steps should be taken to estimate overall cost/benefits of a proposed application:

1) Document the present system operating costs (including such costs as user departmental personnel, SIPS operation personnel, system maintenance personnel, hardware costs, software costs, network costs, special equipment like air conditioning, rents and leases, miscellaneous like stationary.)

2) Develop costs associated with the proposed application. Use the Standard Estimating process prescribed by the IRMC and documented in “IRMC Standard Procedure for Estimating Costs and Delivery Dates” (see OPA section of the SDLC). A “sizing” estimate is usually what is required for a Feasibility Study phase. In general, a sizing estimate has a + or – 75% level of accuracy. Use the IRMC Standard Project Proposal Estimate form to document the estimates.

Evaluate the cost impacts of reusing existing shareable services and the creation of new sharable services that can reduce the cost of other system development projects. For instance, a subscription fee may be the only cost associated with reusing a shared service. Developing a new shared service entails more rigorous and extensive testing and documentation requirements.

Compare the costs of the existing and proposed system. Ask the question, "Could the existing system, with minor improvements, provide the same functionality as the proposed system"? Ask organizational management if benefits could be realized through process/organizational redesign without material system changes. Evaluate the savings in terms of the payback period. How many years would be required before the new system is paid for through savings?

Test and document the logic behind estimates for reasonableness given past experience.

List intangible benefits of the new system and get concurrence from the sponsor and other appropriate management about these benefits. Below is an example intangible benefit:

"An application may lead to fewer, more simplified source documents to process for adding, changing, and deleting personnel information. This should improve the morale of the payroll and personnel people as they were swamped with the detail preparations and filing of poorly designed forms that requested non-pertinent data."

Should the need for a cost/benefit be waived, be sure to get signed documentation to that effect. Waiver of the cost/benefit analysis must be signed off by the ITS Director and the client Division Director and the client Assistant Secretary.

Reference III.D.6 Scope Determination

Develop a formal scope document that identifies the objectives of the project/application, lays out the responsibilities of the project team, specifies the deliverables, identifies the business functions that will be affected, indicates the expected/necessary amount of user involvement, notes the user locations that will be involved, and lays out the business benefits to be achieved.

As the project progresses, it is likely that the scope will also vary. It is important to review any changes in the initial scope of the project with the project sponsor and with other appropriate management. A formal change of scope memo should be developed and signed by the project sponsor whenever the scope is changed.

The initial definition of project scope is subject to revision. As the work plan is developed in more detail, assumptions about project goals and required resources may change. Therefore, outline the scope in as much detail as possible, including all technical and architectural assumptions. Once the final scope and approach are approved at the conclusion of the Workplan segment, plan to control subsequent changes to scope through change request management (change control) procedures. Outline these scope management procedures explicitly in the initial scope memo. It should be noted that scope creep is inevitable and can be constructive as long as it is carefully managed. The core project management team including the database specialist, developers,

project managers, key users, and testers should be part of this team and give their input during the analysis.

Reference III.D.7 Identifying Common Processes

Common Processes must be identified during the Feasibility Study phase. This activity must be performed to support subsequent mandatory IEEE activities (A.2.3 Software Importation Activities) in the next phases of the software life cycle. Common Processes present an opportunity for developing Shared Services or reusing existing ones. There are three levels of commonality that must be considered:

- The process is common within the business unit for which the feasibility study is being conducted (e.g. Access to participant or facilities information).
- The process is common across business units within the same department. For example, different types of Permits or Licenses may be issued by various units with a high degree of similarity in the data to be processed and the steps or activities performed).
- The process is common beyond the department and may also occur in or is performed by other State agencies. An example of this is validating the personal details of an individual applying for a Wildlife Hunting License. In this process, the applicant's driver's license number is obtained and a query is submitted to the Department of Transportation database to obtain personal details such as date of birth and address.

IV – REQUIREMENTS DEFINITION AND CONCEPTUAL DESIGN

IV.A Business Requirements and Conceptual Design Purpose

Provide a broad overview of the proposed application including business requirements, descriptions of high-level functions provided by the proposed application, a high-level representation of data requirements, and a final document that describes the integration of the application with the relevant business process(es).

IV.B Business Requirements and Conceptual Design Objectives

1) Validation and refinement of relevant business process(es).

2) Identification of proposed application business requirements - the "what" needs to be done.

3) Redesign of initial business process(es), where appropriate.

4) Identification of high-level application functional requirements.

5) Identification of reusable software that may be reused or generated by the proposed application.

6) Construction of high-level data model. Object oriented projects construct a class model at this point, the data model will be constructed during the Requirements Specification and Prototype section of the SDLC.

7) Initial development of the Software Quality Assurance Plan.

IV.C Business Requirements and Conceptual Design Outputs

Completing the steps outlined in Section IV.D will result in a Business Requirements and Conceptual Design Report that will define system requirements and design issues necessary to fulfill the project sponsor's needs. The report specifications are defined below:

1) Executive Summary - The executive summary will include an overview of the intermediate documents developed in this phase including: a) a description of the current business process(es), b) a statement of business requirements, c) process redesign recommendations, d) an overview of high-level application functions, e) a reuse proposal and f) description of a high-level data model.

2) Project Background - The project background will include a restatement of the initial request and a summary of the current business process.

3) Process Description – an overview description of the current process and proposed process redesign changes, including anticipated organizational impacts.

4) Business Requirements - This section will identify the business requirements, the system inputs and outputs and their integration requirements. This section should be in outline format.

5) Revised Cost/Benefit Analysis - A revision of the initial cost/benefit estimates in the Feasibility phase.

6) Listing of Proposed Application Functions – This section lists application functions necessary to satisfy the business requirements and implement the redesigned business

process. Reference the business requirements from item 4 by identifying the outline number. Object oriented projects will document the proposed application functions via the actors list and use case models.

7) Reuse Proposal - This section must include a matrix of Shared Services to be Reused or Produced. This identifies how the List of Common Processes (i.e. Reuse Opportunities) is to be transformed (by the proposed project) into actual reuse – either by subscribing to an existing Shared Service, or generating one.

8) High-level Data Model – This section includes the data model diagram and a brief description of the data flow in the redesigned business process. Object oriented projects construct a class model at this point, the data model will be constructed during the Requirements Specification and Prototype section of the SDLC.

9) Initial IRMC Project Proposal Checklist.

10) Signatures - The signature section of this document will be signed by the ITS Director and the project sponsor. A statement should precede the signatures identifying that the ITS Director and the and project sponsor accept the Business Requirements and Conceptual Design Report.

IV.D Steps to Complete the Business Requirements and Conceptual Design Report

Step	Description	Inputs	Outputs	Resources	Notes
1	Establish Business Requirements Design Team			Project Manager/ Project Sponsor	See reference IV.D.1
2	Review Documentation	List of Alternatives with Respective Advantages and Disadvantages, Process Description and Associated Workflow Diagrams, Statement of Problems, Opportunities			
3	Validate and Refine Initial Process Description	Process Description and Any Associated Workflow Diagrams, Statement of Problems, Opportunities	Process Description and Any Associated Workflow Diagrams, Statement of Problems, Opportunities	Team	See reference IV.D.2

4	Define Application Business Requirements	Process Description and Any Associated Workflow Diagrams, Statement of Problems, Opportunities, List of Common Processes	Application Business Requirements	Team	See reference IV.D.3 and reference IV.D.4
5	Where appropriate redesign the original process to enhance productivity, enable cost savings, and/or improve customer service.	Final Process Description and Any Workflow Diagram, Refined Statement of Problems, Opportunities, Application Business Requirements	Recommended Process Changes, Process redesign matrices	Team	See reference IV.D.5
6	Determine High-level Application Functionality	Recommended Process Changes, Process redesign matrices, Final Process Description and Workflow Diagram, Business Requirements	High-level Application Functionality Document	Team	See reference IV.D.6,
7	Analyze reuse opportunities	List of Common Processes Reuse Candidates from DENR or IRM Component Repository	Reuse Proposal	Team	See reference IV.D.7
8	Develop High Level Data Model	High-level Application Functionality Document, Recommended Process Changes, Process redesign matrices, Final Process Description and Workflow Diagram, Business Requirements	High Level Data Model	Team/DBA	See reference IV.D.8

9	Review and revise initial cost and risk estimates	Original cost/benefit/risk analysis Final Process Description and Workflow Diagram, Business Requirements, IRMC Procedure for Estimating Costs and Delivery Dates	Revised cost and risk estimates	Project Manager/ Team	See Reference IV.D.9
10	Prepare Business Requirements and Conceptual Design Report	Results from Previous Steps	Draft Business Requirements and Conceptual Design Report	Team	
11	Prepare initial IRMC review requirements	FORMS: Project proposal checklist (PPD), Technical Capability Table (TCT), Project Concept Document Template	Draft IRMC Project Concept Document	Team	See Reference IV.D.10
12	Select Internal Quality Assurance Team		Internal Quality Assurance Team Roster	Project Manager, Project Sponsor	See reference IV.D.11
13	Internal QA Review	Draft IRM Review documents and Initial QA Plan for pre-development phase.	Internal QA Review Comments, Initial QA Plan	Internal QA Team	See reference IV.D.11
14	Incorporate Internal Quality Assurance Comments	Internal QA Review Comments	Final Business Requirements and Conceptual Design Report	Team	
15	Obtain Approval and Commitment to Continue	Business Requirements and Conceptual Design Report	Signed Business Requirements and Conceptual Design Report	Team, Project Sponsor	

References

IV.D.1 Establish Business Requirements Team

As with the Feasibility Study Team, this team should, at a minimum include the project sponsor and project manager. In addition, the key focus in this section is the identification of business requirements, the things the system must do to meet the needs of the business. User involvement helps ensure that the system will satisfy management expectations. In the long run, serious problems, rework, conflicts and delays may be prevented by the involvement and active participation of qualified, empowered users. User involvement in the system development process facilitates acceptance at an early stage; helps to champion and gain support for the project throughout the organization; maintains a realistic level of expectations; and ensures that the true business requirements are being addressed. Some effective techniques for involving end users in specification and design of systems are joint application development (JAD), focus groups, facilitated meetings, prototyping, usability tests, and participation in issue and business case analysis.

A representative from the IRM should be invited to participate in this phase.

It is critical that all team members work in or near the same location-especially during analysis and design, when an extensive amount of interaction is needed.

IV.D.2 Validate and Refine Initial Process Description

Validation and refinement of the initial business process description usually requires interviews and walk throughs with operational users and middle management. Often many operational users perform the same work. When this is so, select a representative for the group(s) that performs common tasks. It is generally useful to discuss the selection with the manager of the relevant workgroup to identify a preferred candidate. When selecting interviewees, keep in mind the need to build consensus and ownership for the new system.

When conducting the interviews and walkthroughs note operational characteristics like resources required, backlogs, bottlenecks, incidents, and complaints. Also maintain a list of issues and problems that were not identified initially and should be resolved.

IV.D.3 Define Business Requirements

Consider the following questions when identifying business requirements:

1) What are the important problems/objectives as they relate to the business process? (e.g. The legislature is complaining that our permitting application process is too slow. They have received numerous complaints from their constituents.)
2) What are the critical success factors? (e.g. We must receive reliable and comprehensive information from the applicant during the initial application.)
3) What are critical assumptions? (e.g. An implicit current assumption may be that reviews by staff must be sequential. A new assumption may be that at least some of the reviews may be conducted concurrently.)
4) Distinguish the requirements of the key stakeholders groups: executive/sponsor, supervisory/controller, and operational/user.

11) Distinguish between true requirements and “nice to haves.”
12) When identifying the business requirements ask, “What are the business rules (e.g. these positions must sign off), activities (e.g. an initial letter that acknowledges receipt of an application must be posted), and data (e.g. the pollution source) that make up the relevant business process and are necessary to satisfy the business requirements?”
13) What other state agencies need this functionality or access to this data? What other state agencies

might already provide this functionality or maintain this data?

IV.D.4 Implementation Requirements

Implementation Requirements indicate how the application system must operate or perform to meet the business requirements. At a minimum the following high-level implementation requirements should be considered:

1) Extent to which system processing requires management authorization.
2) Extent to which system activity must be traced or audited.
3) Required degree of system reliability and continuity of systems operations, and conversely, implications of system failure.
4) Extent to which schedules must be met (e.g. payroll check distribution).
5) Extent to which system access must be restricted.
6) Ease of use.
7) Effort required to locate and fix error in operational programs.
8) Extent to which systems can operate in multiple environments now and in the future.
9) Extent to which system must integrate with other systems.
10) Extent to which system can survive external breakdowns (e.g. redundancy of components and communications paths.)
11) Extent to which functions/components can be changed rapidly (e.g. required modularity.)
12) Extent to which functions/components can be reused.
13) Extent to which system can scale for increased workloads.

IV.D.5 Process Redesign

To redesign the process consider the following steps:

1) To understand the process structure, review the validated workflow diagram recognizing activities, steps, inputs, outputs, and triggers (assuming the business process is consistent with a workflow presentation.) Then ask the following questions: a) What organization and jobs are involved in the process? What pieces of work are done by each job? b) What policies/business rules apply to the performance of the process? In which piece of work does each policy/rule apply? c) What technology is used in the process? In which piece of the work is the technology used?

2) To understand the process flow, review the validated workflow diagram recognizing activities, steps, physical (e.g. customers, employees, products) and transactional (e.g. orders, invoices) inputs and outputs, and triggers. Object oriented projects would review use case models. Then ask the following questions: a) What are the important decision points in the process and how do decisions determine the path activities/steps taken? b) What inputs and outputs are related to each activity/step and how are the inputs manipulated or transformed (e.g. created, updated, read)? The relationship between inputs/outputs and activities or steps is best represented in a matrix. c) What time is required to execute each activity/step? Assign time to each activity step. (When assigning time consider "in-que, in-process, out-que" times for each activity/step") (See Appendix I for example of workflow diagram and process flow matrix.), d) Determine which activities/steps are value-adding (i.e. directly benefit customers), for internal control, and other.

3) To identify the opportunities for process redesign including changes in inputs, outputs, and processes, review the process structure and flow description and matrices. Look for opportunities to minimize/streamline control activities/steps. Evaluate each activity/step looking for the: a) opportunity, b) change requirements, c) difficulty, d) benefits, e) costs, f) support requirements, and g) risks. Also reexamine process linkage changes (e.g. rearranging steps within an activity or activities within the process.) Look for opportunities to add steps that measure process performance by capturing, assembling, and distributing performance information (e.g. cycle time, accuracy, reliability.)

IV.D. 6 Determine High-level Application Functionality

To determine the proposed application system functionality, review the business requirements and the activities/steps of the redesigned process. During this review consider the following general functions and characteristics that can benefit from technology solutions:

Analysis, e.g. simulations, trends, variance analysis
Capture and storage, e.g. imaging, data entry, satellite transmission and GPS
Communication, e.g. ftp, email, web page
Control, e.g. system checks, automated alerts, automated routing
Human interface, e.g. GUI, pen-based, voice recognition
Identification, e.g. bar codes, transponders, magnetic stripes
Management, e.g. decision support
Relationship with other application systems and sharable services

IV.D. 7 Develop Reuse Proposal

Using the List of Common Processes, evaluate how the requirements of these processes may be satisfied:

a) Determine if the Common Process is supported by existing software either as a whole or in parts. Check with the ITS SDT Manager for a list or database of DENR Shared Services. The IRM is implementing a component repository that stores information about Statewide Shared Services – check with the IRM if this is already available.

   Within the context of the NC Statewide Technical Architecture, a "Shared Service" or reusable component refers to an *executable* program module that is accessed through the NC Service Broker middleware backbone. (This is a more limited definition of reusable components compared to other organizations where "components" may include other objects such as design components, pre-developed architecture, test scripts, etc.). Both the Shared Service and the program accessing the service must use the Service Broker Application Programming Interface (API).

b) Determine the feasibility of using the existing Shared Service. Determine if an expansion of the Shared Service is necessary. Expanding the functionality of a Shared Service should be carefully considered since Shared Services (i) should be as generic as possible and customization may impact other users of the Shared Service, and (ii) according to NASA, revising more than 25% of a reusable software will cost as much as if the software were built as a new module.

c) If the Common Process is automated – perhaps within a legacy application - but is not currently accessible to other applications, determine the feasibility of "harvesting" the legacy component to become a Shared Service by wrappering the Service Broker API around it.

d) If the Common Process is not automated, subsequent development must ensure that the corresponding software is designed to be a Shared Service.

In the case of c) and d) above, the project must identify and include in the Requirements Definition process, other parties that employ the common process (within and outside of the Department).

Create a matrix showing the Common Process and whether an application function will be reused, harvested, or generated as a Shared Service by the project.

IV.D. 8 Develop High-level Data Model or Class Model

To build a high-level data model you will be identifying data entities (things or objects that make up the world, things the organization must keep data about to conduct "business") and their relationships (i.e. one to one, one to many, or many to many) with one another. Refer to the process redesign materials, with

particular attention to inputs and outputs. In addition, consider the following "entity types": a) people (e.g. employee, customers), b) places (e.g. offices, regions, locations), c) organizations (e.g. agencies, vendors), d) concepts (e.g. orders, projects, accounts, invoices, applications, permits), and e) events (e.g. inspections, license issuance). You may want to bring in an expert to assist with the data modeling effort (e.g. SIPS DBA.) Object oriented projects capture this information by building a class model. When constructing this model consider the following:

- Other divisions/sections/external agencies who may also have an interest in the data being collected for and processed by this application. Contact these organizations and request an end user participant where appropriate.
- Complying with the State Technical Architecture requirements for federated data (i.e. data must be consistently defined, shareable, accurate, centrally managed).
- Adopting IRMC–approved statewide federated data standards (e.g., name, address, date, etc).
- Also assuming that the department has proceeded with the IT Planning recommendation to establish a corporate data warehouse, contact the departmental data administrator and request that he or she participate in the modeling exercise. Aside from technical assistance, this person should focus on data requirements that should be incorporated into the departmental data warehouse.

IV.D.9 Refine Estimates

Refine the original "sizing" estimate using the IRMC-prescribed process for estimating. At this stage, an "approximation" is expected – i.e. an estimate with a + or – 25% degree of accuracy. Note that further refinement may be done at the Architecture segment of the software life cycle.

IV.D.10 IRMC Requirements

During this segment, the following documentation should be reviewed, considered during planning and analysis, and drafted:

- Project Proposal Checklist (PPC) – the PPC evolves as the project goes through the software and project life cycle. Project teams are not expected to have all the checklist questions answered at this time, but an indication of when and how the items will be addressed is required. The PPC is available at the IRMC website. See Appendix for website address.

- Technical Capability Table (TCT) – originally intended for vendors proposing technological solutions to State agencies, this table contains all the applicable STA standards and may be used by agencies as a tool to assess and/or demonstrate compliance with the Statewide Technical Architecture. The TCT is available at the IRMC website. See Appendix for website address.

- Project Concept – the Project Concept is an overview document that summarizes the information specified in the Business Requirements & Conceptual Design Report (what, why), identifies who are involved in the project (who), and identifies the project resource requirements. This is the main vehicle to be used when applying for IRMC project certification. . A Project Concept template is available at the IRMC website. See Appendix for website address.

These documents must be finalized at the end of the Architecture segment of the software life cycle process, and submitted as a packet to the IRM to initiate the IRMC project review, approval and certification process.

IV.D.11 Quality Assurance

Quality Assurance must be applied on application development projects.

- Internal Quality Assurance Reviews

These are QA activities that are equivalent to IEEE-defined "Reviews" (see Activity A.5.1.1 under IEEE Standard 1074-1997). These may include walkthroughs, peer reviews, inspections, management review, and process improvement reviews. These activities are performed by the project team.

A quality assurance plan is the basis for conducting QA Reviews. Within the Business Requirements & Conceptual Design segment of the software life cycle process, a subset of the Software Quality Assurance Plan (SQAP) should be developed to address relevant products to be produced and processes to be used during the Pre-Development Phases. For example, the Internal QA Review in this segment of the software life cycle could be in the form of a Walkthrough of the draft Project Concept document and a review of the supporting documents.

A full-blown SQAP is required at the beginning of the Requirements Specifications segment of the software life cycle process. Use the prescribed IEEE Standard for Developing Software Quality Assurance Plans (#730-1998) - see the Appendix for file location.

- External Quality Assurance Review

External QA Reviews correspond to the IEEE-defined "Audits" (see Activity A.5.1.3 under IEEE Standard 1074-1997). Audits are performed by an independent party who is not part of the project.

Third party QA reviews are mandated by the IRMC for projects that meet specified threshold levels. The IRM QA group schedules at least 3 reviews: initial, in-process, post implementation. IRMC QA Review requirements are published at the IRMC website – see Appendix for website address.

V - ARCHITECTURE

V.A Architecture Purpose

Identify the most appropriate technical and logistical system architecture for proposed business process/business requirements implementation.

V.B Architecture Objectives

1) Identification of architecture selection criteria.

2) Selection of preferred application system architecture with documentation of the selection rationale.

3) Initiate IRMC Certification process.

V.C Architecture Outputs

Completing the steps outlined in Section V.D will result in the Architecture Report and completing the steps outlined in V.E will result in project certification or non-certification by the IRMC.

The Architecture report specifications are defined below:

1) Executive Summary - The executive summary contains an overview of the selection criteria, the architecture selected and the rationale behind the architecture.

2) Selection Methodology - The selection methodology includes a list of the selection criteria and their rank and weight. The selection methodology must also include a description of the reasoning behind the selection of the criteria.

3) Alternatives - The alternatives should include a brief description, advantages, and disadvantages of all the architecture options considered.

4) Recommendation - The recommendation will provide details on the architecture selected. The recommendation must also contain the rationale for the selection, and a Traceability matrix showing how each of the Business Requirements is addressed by the recommended architecture.

5) Reviews - The reviews will contain comments made by the Quality Assurance Team and any other teams required (e.g. TAC~~, IRMC~~).

6) Signatures - The signature section of this document will be signed by the Director of ITS and the project sponsor. A statement should precede the signatures identifying that the ITS Director and project sponsor accept the Architecture Report.

V.D Steps to Complete Architecture Report

Step	Description	Inputs	Output	Resources	Notes
1	Define the Architecture Team		Team Roster including Project Manager, Technical Experts, Process Experts	Project Manager and Sponsor	See reference V.D.1
2	Review Documentation	Feasibility Study, Business Reqts & Conceptual Design		Team	
3	Choose Architecture Selection Criteria and Determine the Rank and Weight of the Selected Criteria	Business Requirements, Technology Trends, NC Statewide Technical Architecture, List of Architecture Selection Criteria	The Architecture Selection Criteria and Rank, Weight and Rationale for Each	Team	See reference V.D.2
4	Identify Alternative Architectures	Business Requirements, Functional Requirements, Data Requirements, Technical Reviews and Specifications , Statewide Technical Architecture	List and Description of Alternative Architectures	Team	See reference V.D.3
5	Compare Alternative Architectures Based on Selected Criteria	List and Description of Alternative Architectures, The Architecture Selection Criteria	Architecture Alternatives with Criteria Rankings	Team	Several Iterations May be Required
6	Determine Architecture Recommendation	Architecture Alternatives with Criteria Rankings	Architecture Recommendation, and Traceability Matrix	Team	See reference V.D.4
7	Prepare Architecture Report	Architecture Recommendation , and Traceability Matrix	Draft Architecture Report	Team	

Step	Description	Inputs	Output	Resources	Notes
8	Internal Quality Assurance Review	List and Description of Alternative Architectures, The Architecture Selection Criteria, Architecture Alternatives with Criteria Rankings	Internal QA Comments	Internal QA Team	
9	Incorporate Quality Assurance Comments	Internal Quality Assurance Comments	Architecture Report	Team	
10	Obtain Senior Management Approval and Commitment to Continue	Architecture Report	Signed Architecture Report	Team, ITS Director, Project Sponsor	

References

V.D.1 The Architecture Team

The team may include ITS-Commerce/IRM or External Consultants. Composition may change as this phase progresses and different expertise is required.

Ensure that there is IRM representation in the Architecture Team or that there is in-process communication with an identified IRM resource to validate the direction of the proposed architecture as the team is shaping it.

V.D.2 Architecture Selection Criteria

Consider the following list of potential criteria. Select and rank those that pertain to this application and its related project.

1) To what extent does the proposed system support the requirements or adhere to the standards specified by the NC Statewide technical architecture? Use the Technical Capability Table as a tool to assess compliance – see Appendix for file location.
2) Is the business function centralized or distributed?
3) Is this a new application or an enhancement to an existing application?
4) What skills are available to build the application?
5) Is there ~~an~~ organizational (e.g. State, Agency) strategy or standard that applies?
6) Portability (Will this system need to function on multiple platforms)?
7) Scalability (Will this system need to function on small, mid-size, and large platforms)?
8) Common tools. Are the selected tools compliant with the Department standards? If not, has the non-compliance been endorsed by the DENR IRMB?
9) GUI (Does this system require a graphical user interface, and should this interface have the same "look and feel" as other systems)?
10) Security (What are the security requirements for data within this system)?
11) Useful life (Is this system a short term fix to be replaced in the near future or will it remain in use for many years)?
12) Cost (Are there funding limitations that will eliminate options that otherwise would have been

preferred)?

13) Future rate of change (Will there be few or many changes to this system after implementation. A high rate of change may favor a centralized system where change is more easily managed)?
14) Input sources (What are the form, format, volume, and frequency of inputs)?
15) Output destinations (What are the form, format, volume, and frequency of outputs)?
16) Deadline scheduling (Are there implementation deadlines that will eliminate options that otherwise would have been preferred)
17) Emerging technologies (Are there new or anticipated technologies that may apply to this system)?
18) Industry standards or practices (e.g. EFT, EDI, SMTP)?
19) Type and skills of the users?
20) If client, what servers will it interface with?
21) If server, what clients will it interface with?
22) Buy or build?
23) Synergy with other applications (Is there rationale for this system to utilize the same architecture as other systems; common code, common platforms, common user interface, etc.)?
24) Communications interfaces (What are the form, format, volume, and frequency of communications with other systems)?
25) What platform (hardware and software)?
26) Sole use of platform or shared?
27) Vendor market share (Without significant market share there is little assurance that the technology will remain current and be supported by the vendor in the future)?
28) Vendor financial stability (Without financial soundness there is little assurance that the technology will remain current and be supported by the vendor in the future)?
29) Performance requirements?
30) Reliability requirements?
30) Availability requirements?
31) Functional requirements?
32) Inter-operability?
33) Maintenance Requirements?

V.D.3 Architecture Selection Considerations

Architecture Concerns:

1) user interface approach
2) application distribution approach
3) data distribution approach
4) networking approach
5) external systems interfaces approach
6) operations management approach

Platform Concerns:

1) workstations and servers
2) user interface environment
3) data management software
4) programming language(s)
5) development environment and CASE tools

Highlight any architecture and platform areas that cannot be identified, and schedule them for resolution as early as possible. One area that cannot be deferred, however, is the workstation, local area network (LAN) and software that constitute the analysis and design environment - particularly the prototyping tool. These choices are on the critical path for all the analysis activities. If the user interface environment is not selected as part of the feasibility design, identify an interim prototyping approach and tool and make a final selection during this phase.

V.D.4 Determine Architecture Recommendation

The final recommendation should include a number of checks including:

1) architectural features should be mapped against user requirements to be sure all requirements are supplied (use the Traceability Matrix),
2) architectural features should be mapped against quality requirements to ensure that performance, reliability, usability, and flexibility requirements are supported (use the Traceability Matrix),
3) architectural features should be mapped against the Statewide Technical Architecture (use the Technical Capability Table).
4) demonstrate how the use of Shared Services is to be supported (both types: shared services reused or produced),
5) all changes and agreements have been documented,
6) the recommended architecture capacity meets the needs of the proposed application,
7) the technical architecture is developed in sufficient detail for application developers to begin technical design, and
8) the project management team will be able to secure sufficient resources to test and build the technical architecture.

V.E Steps to Obtain Project Approval from the IRMC

1	Review project documents and Prepare IRMC-required review and approval documents	Business Requirements & Conceptual Design Report, Architecture Report, Project Proposal Checklist, Project Concept document	Update Project Concept Document, Project Proposal Checklist	Project Manager	
2	Submit to Project Sponsor and ITS Director for Approval	Updated Project Concept Document, Project Proposal Checklist (PPC)	Signed Project Proposal Checklist	ITS Director and Project Sponsor	
3	Initiate IRMC certification process	Signed PPC, Project Concept Document, Architecture Report	Notice of certification or non-certification	IRM CTO, IRMC	See IRMC website for submission procedures.

The Project Concept document template, and the Project Proposal Checklist (PPC) may be obtained from the IRMC website – see Appendix for website address.

VI - REQUIREMENTS SPECIFICATION AND PROTOTYPE

VI.A Requirements Specification and Prototype Purpose

Requirements specification refers to the refinement of original high-level functions and high-level data requirements. The final specifications at the system level define the required behavior and properties of the system. At the system design level, specifications define how the various components including hardware, software, displays, networks, operators, etc. are to be integrated. Finally, at the software level, the required behavior and properties of software components are specified.

VI.B Requirement Specifications and Prototype Objectives

1) Definition of detailed function specifications.

2) Identification of function outputs and inputs.

3) Definition of processing requirements.

4) Development of prototype (if appropriate.)

5) Update of Resources/Benefits associated with the proposed application and related project.

6) Completion of the Software Quality Assurance Plan.

VI.C Phase Outputs

This segment of the software life cycle will produce the following documentation:

- Final System Report Output
- Software Quality Assurance Plan

The Software Quality Assurance Plan (SQAP) must be developed based on the IEEE Standards for Software Quality Assurance Plans (#730-1998).

The Final System Report will describe all the required specifications. It also updates the anticipated benefits of the proposed application and resource requirement associated with the related development project. For relatively small and/or poorly defined applications, iterative prototypes may be developed to clarify requirements and expedite the design/development process. The sections of the report are described below:

1) Executive Summary - The executive summary provides an overview of the rationale behind the proposed system and the related development project, the objectives and scope of the proposed system, an overview of the application specifications.

2) Description of the Current System (if one exists): a brief description of the current system, its purpose, and any problems.

3) Description of the Proposed System: This description will detail all functions being implemented by phase, including inputs, outputs, processing requirements, and methods for testing functionality by phase. Object oriented projects capture this information by producing an actors list, use case model, interaction model, physical system architecture and physical data model. Any reservations about the timelines, staffing, hardware, software, etc. should also be discussed.

4) Resources/Benefits Associated with Proposed System: The resources represented in this section include a refined estimate of personnel needs, development, and operating costs of the new system. Benefits are both tangible and intangible.

5) Signatures - The signature section of this document will be signed by the ITS Director and the project sponsor. A statement should precede the signatures identifying that the ITS Director and project sponsor accept the Final Systems Report.

Note: Specifications derived from prototyping would be included in the "Description of the Proposed System" section of the Report. The project sponsor would have access to and approve the prototype before the Final Systems Report is signed.

VI.D Steps to Complete the Final System Report and Prototype

Step	Description	Inputs	Outputs	Resources	Notes
1	Assign Project Management Team members to the Requirements Specification and Prototyping Task		Specifications and Prototyping Team	Project Manager & Project Sponsor	See reference VI.D.1
2	Review Documentation	Feasibility Study, Requirements Definition and Conceptual Design, Technical Architecture Report, Work Plan		Specifica-tions and Prototypi ng Team	
3	Refine Functional Requirements for Each Phase	Requirements Definition and Conceptual Design, Technical Architecture Report, Work Plan	Draft of Refined Functional Requirements for Each Phase	Specifica-tions and Prototypi ng Team	See reference VI.D.2, VI.D.3
4	Determine Required Function Outputs	Draft of Refined Functional Requirements for Each Phase	Draft of Function Outputs	Specifica-tions and Prototypi ng Team	Outputs requirements based on user reports or displays
5	Determine Detailed Data Requirements Based on Outputs and High-level Data Model	Results from Step 3 & 4	Detailed Data Requirements	Specifica-tions and Prototypi ng Team	See reference VI.D.4
6	Determine the Processing Requirements	Results from Steps 3, 4 & 5	Draft of Processing Requirements	Specifica-tions and Prototypi ng Team	

Step	Description	Inputs	Outputs	Resources	Notes
7	Create Prototype (if requirements are poorly defined and project is relatively small.)	Requirements Definition and Conceptual Design, Business User Discussions	Draft Prototype	Specifica-tions and Prototypi ng Team	See reference VI.D.5
8	Review/Accept-ance of Prototype	Results from Step 7	If Project Sponsor Accepts Prototype, go to Step 9, Else go to Step 3.	Project Manager, Project Sponsor	Steps 3 - 8 Should be Repeated Until the Prototype is Accepted.
9	Internal Quality Assurance Review	All Business Requirements, Functionality, and Data Modeling Documents, SQAP	Quality Assurance Comments	Quality Assuranc e Team	
10	Prepare Final Systems Report	Results from Previous Steps	Final Systems Report (including Reqt Specs and Prototype Report)	Specifica-tions and Prototypi ng Team	See reference VI.D.6
11	Obtain Approval and Commitment to Continue	Final Systems Report	Signed Final Systems Report	Team, ITS Director, Project Sponsor	

References

VI.D.1 Assignment and Staffing

Two different approaches to staffing this segment are possible. These might be referred to as the expert and the novice approaches. In the expert approach, experts in the functionality of the application make up most of the team. The advantage is that they can very quickly establish rapport with the users, understanding and documenting requirements and issues efficiently. The team can remain quite small, typically two or three experts, including the project manager, plus clerical support. This approach is preferred when the qualified personnel are available.

The second approach, putting together a team of people who know little about the user's domain, is easier to staff, but takes longer to execute. There is also a significant risk that the users will fail to develop faith in the ability of the team to create a system responsive to their needs. A novice team must be larger than an expert team (although it should never exceed 10 people; if it does, the scope is too wide and the project should be split). The more that change to the work flow is contemplated, the riskier it is to determine requirements with a team of inexperienced or junior personnel. The risk can be mitigated by frequent peer and management reviews or, if an expert is available part time, expert consultations.

If the project requires reverse-engineering the existing system, a larger team of less experienced personnel can do so, especially if automated reverse-engineering tools aid them.

If JAD sessions are used, a qualified session leader must be identified. Session leaders are required to have specific qualifications. They must quickly gain the respect of management, users and the IS team; they must be effective negotiators; and they must have experience in facilitating meetings. Session leaders should also be independent; they should be in the line of command of neither the users nor the IS participants.

For one-on-one interviews, count one hour of preparation and two hours of debriefing for every hour of interview.

For each day of the JAD session, count five to eight work days for preparing and five to eight workdays for debriefing. Do not include too many users in a JAD session; five to eight is ideal.

VI.D.2 General System Design guidelines

The general guideline to follow is to develop the models and the prototype to a sufficient level of detail that the users can validly sign off on the user specifications report - they have understood it and agreed to it. What this implies for the exact structures and objects that the models must contain varies from project to project - the sophistication of the users, the experience of the project team, and the degree of innovation in the application. The following checklist of exit conditions can be used to establish the proper level of detail:

Are all business functions and significant events described with their responses? Are volume, frequency and response time information documented in the model?

Are all transactions and events that do not result in an immediate response decomposed at least one level down?

Are business functions and processes decomposed down to the level of a business unit of work" Are complex processing rules spelled out?

Are all entities and relationships identified? Is volume information documented in the model? Is the behavior of entities over time understood? Are enough attributes identified that users understand what information will be kept about the various entities and relationships?

Does the level of prototyping chosen correspond to the intended degree of innovation in technology, workflow and functionality? Are the functions and features prototyped a representative sample of the future application? Have issues raised during prototyping been fed back to the team and integrated into the requirements specification? Can all the components of the requirements specification and the prototype be justified by their contribution to the business case? Have any special test requirements been identified and documented? Have the relevant acceptance criteria been defined and documented?

Is it clear how these designs contribute to the overall requirement? Have all disagreements with stakeholder groups, between the stakeholders and project team or with other requirements owners been managed to a satisfactory conclusion?

Where any of these questions remain unanswered, the team should ensure that further requirements capture and analysis is carried out when appropriate.

VI.D.3 Refine Functional Requirements

Joint Application Development (JAD) sessions represent an acceptable approach for requirements gathering, verification or planning activities. Joint Application Development is applicable to situations that require input from multiple user groups and group consensus. In general the following guidelines indicate the applicability of JAD:

Traditional approaches to requirements gathering or other user interaction have not produced acceptable results
Project will impact multiple groups
Users play different roles and have different levels across organizations
The relationship between IS and the user community requires strengthening
User acceptance and/or commitment is an issue

JAD sessions do not perform or replace the work performed by the project team. Rather the JAD session provides user input and verification in a single session as opposed to a series of interviews or meetings.

The JAD process consists of Plan, Conduct and Review. Each of these steps is required to have a successful JAD session. Several types of personnel are involved in this process including: Project Management, Executive Sponsors, IS Personnel, Users, a recorder and a facilitator.

The facilitator is one of the key people in the JAD framework. They are responsible for planning and conducting the JAD session. In this capacity they are central to the participants defining the requirements and/or designs of the new system. The facilitator should have the following characteristics:

1) Independence from the IS and user organizations

2) Possess industry knowledge

3) Experienced in group meeting facilitation and discussion techniques

Resist the temptation to staff a member of either the IS organization of the user groups in the role of facilitator. The facilitator should be an honest broker among the various groups of JAD participants. This need to remain impartial is essential to achieving user consensus and support for the JAD session.

Planning is essential for a successful JAD session. The planning and preparation work establishes the scope, objectives and logistics for the session. The project sponsor approves and is involved with all facets of the planning activities. The plan should include identification of the participant list for the session. Additional considerations addressed in planning include:

Scope of the JAD session and the system it will discuss, this is required in order to provide a boundary for the session. Unbounded discussions are difficult to control and do not produce the desired results.

Objectives of the JAD session, the objectives create an organization to the session's materials and topics. Activities, the scope and objectives of the JAD session should define a set of outputs from the session. These outputs may be created through activities during the JAD session.

Participant materials give the attendees sufficient information to prepare for the JAD session. Attendee preparation increases the quality of their contribution and the results of the session.

Location of the session, the JAD sessions should be located to remove the user from daily distractions.

Length of the JAD session, this is dependent upon the scope and objectives in the session. Generally individual session may run from one to three days. An entire JAD process may span several sessions.

Format or scope for the JAD session, the scope addressed in the session will help determine who should attend and how long the session will be.

Attendees at the session, the attendees provide the definitions and input to the systems building process.

VI.D.4 Determine Detailed Data Requirements

Required input-related information include file specifications including all data elements and their attributes, an index of table field names, and the input layouts (i.e. the format of all data to be captured by the system.). In addition, data distribution guidelines should also be established and approved including guidelines for data ownership, data access, concurrency of access, regulatory constraints, projected volumes, database recovery and restart, data sharing with other systems, and data integrity. These guidelines need to address the ownership of both shared data and services.

In some cases, data entities that are of interest to other organizations will be part of the high-level data model. If this is the case, be sure to include representatives from these organizations when determining detailed input requirements. These representatives should play a key role in specifying the detailed requirements associated with the data entities that are of interest to them.

VI.D.5 Create Prototype

A prototype is a model of the application prior to its creation. The prototype is based on partial completion of the analysis models and/or requirements. Prototypes serve two general purposes in regards to requirements: exploration and verification. In many cases, particularly where the application is not well-defined, this model is useful. However, it is sometimes not worth the effort required to develop the model in which case you may choose to skip the prototype related steps.

The precise sequence of steps to construct a prototype depends on the tool used. For the selected functions, the finished prototype should accomplish all of the following:

1) Illustrates the layout of all the window types and widgets (look)

2) Allows the user to navigate from window to window (feel)

3) Allows the user to enter data into the system and select items from list boxes uses all the input devices that will be a regular part of the system

Test the prototype to ensure that it will perform as expected. Neither the coding nor the testing will be as thorough as for the final application. If you find errors that are hard to correct (or if the prototyping tool does not support certain functions that will be supported in the production environment), document these limitations and clarify to the users that they do not reflect the production application.

Limit the scope of the prototype to a subset of representative tasks and dialogs to reduce the work without sacrificing any of the benefits.

If several different aspects of the application warrant prototyping, consider creating multiple prototypes, one for each area. The complexity and cost of prototyping can easily become excessive if the prototype attempts to do too much. Do not expect the prototype to evolve into a working application.

Since the main goal of a prototype is to gauge the users' reaction to the future application, develop several sections of the application to provide data entry and retrieval capability, full navigation, etc. The sections of the application to prototype with full-function appearance are those that are frequently accessed by the user, characteristic of many different sections of the application, or particularly complex from the users' standpoint.

The performance of the prototype should indicate the performance of the future system. If the prototype performs too efficiently, the users will expect similar performance when the production system is running. In fact, while users interact with the prototype, check their tolerance of the system response time that was estimated during the analysis phase. Different delays can be used to evaluate user productivity and satisfaction. Other usability attributes can also be tested in the prototype, such as different levels of user guidance, multiple language support, and context maintenance.

VI.D.6 Final Systems Report

The Final Systems Report's Description of the Proposed System must include the following:

1. Requirements Specification and Prototype Documentation – see the Documentation Guide Section 7.B and Section 7.F (object-oriented deliverables).

2. Refined cost and effort estimates. The project estimates should be refined and the updated estimates documented in this segment of the software life cycle. Use the IRMC Standard and Procedure for Estimating Costs and Delivery Dates. Estimates at this point should have a greater degree (+ or – 10%) of accuracy.

3. Reuse method. The method of access for reusable components must be documented. The following must be considered:

- Existing statewide Shared Services are accessible through the State's Service Broker. Therefore, program modules requesting service from Shared Services must use the NC Service Broker (NCSB) API and be designed for asynchronous message processing.

- Shared Services that the new application will produce must likewise be accessible through the Service Broker and adhere to standards that may be defined for statewide or department-wide shared services. Current Service Broker messaging standards exist and are published at the NCSB website. Consult the NCSB support group for more information about using the NCSB infrastructure for shared services development.

VII - PROGRAM DEVELOPMENT

VII.A Program Development Purpose

To write the detailed program code for the system.

VII.B Program Development Objectives

1) Where program development is being performed by multiple programmers, assign the modules to various programmers and track the assignment.

2) Based on the Specifications in the Final System Report, write the detailed code.

34) Determine the distribution of code modules across available servers.

4) Test the modules to assure proper functionality.

5) Prepare individual module documentation.

6) Periodically perform code reviews on individual modules.

VII.C Program Development Output

Completing the steps outlined in Section VII.D will result in a Program Development Report and a completely coded module. The report specifications are defined below:

1) Program Development Report - The program development report should provide detailed information on each module coded. This information will include the purpose of the module, the tables used, a list of the modules that call the current module (plus parameters passed in), a list of the modules that the current module calls (plus parameters passed out), what the module needs to run (VBXs, Crystal Reports, etc) and any other pertinent comments. Object oriented projects should provide detailed information on each class coded. This information will include any inheritance used and descriptions of the methods implemented in the classes.

The code module specifications are defined below:

1) A module that has all the necessary functionality provided in the Requirements Specification and Prototype Report. The module should also accept all inputs and produce all outputs as outlined in the report.

Additionally, a Distributed System Architecture document will detail the location of each code module being deployed as part of the production system.

VII.D Steps to Complete the Program Development Report and Coded Module

Step	Description	Inputs	Outputs	Resources	Notes
1	Review Final System Report and Related Documentation	Final System Report and Related Documentation		Project Manager/ Programming Staff	See reference VII.D.1
2	Assign Modules to Programmers	Requirements Specification and Prototype Report	Module Assignment	Project Manager	Assignments Need to be Tracked. Steps 2 - 4 Should be Repeated Until the Module is Completed. See reference VII.D.2
3	Write Detailed Code	Requirements Specification and Prototype Report	Coded Module	Programmer	
4	Test Detailed Code	Coded Module	Tested Module	Programmer	See reference VII.D.3
5	Write Module Documentation	Results from Step 3 and Requirements Specification and Prototype Report	Module Documentation	Programmer	Steps 2 - 4 Should be Repeated Until the Phase is Completed.
6	Code Review	Results from Steps 3 & 4		Code Reviewer and Programmer	Code Reviews Should Happen Periodically Throughout the Coding Process
7	Finalize Program Development Report	Results from Step 4	Program Develop-ment Report	Project Manager	

References

VII.D.1 Review of Final System Report and Related Documentation

This step is to ensure that there are no misunderstandings of specifications requirements between the systems analyst or user and the programmers. The review should include: a) the larger specifications logic, how the specifications would logically relate to one another, and b) a detailed look at the inputs and outputs at the data element/attribute level.

VII.D.2 Assign Modules to Programmers

Develop the programming schedule such that the assignment and ordering of programming work units are coordinated with the way programs will be unit and system tested. Be sure all dependencies have been considered and any resulting contingency plans have also been developed.

VII.D.3 Code and Testing - Client/Server

Client/server software typically contains many application components across multiple hardware platforms; carefully manage the version and location of all the required components. As code progresses through the systems development process - from coding, through the stages of testing, to production - be sure the configuration is complete and synchronized at each stage. If it is not, the testing process and the integrity of the production system could be harmed.

The Statewide Technical Architecture (STA) identifies N-tier application architecture as the standard architecture for new State applications. This approach specifies designing granular applications where each of the application's program modules is bounded by a business event. In other words, the logical boundaries for application modules are drawn around units of work (i.e. Each business event like "issuing a check" is identified as is the unit of work servicing that event.) Each unit of work is implemented as a collection of related business rules. "Highly granular" and "loosely coupled" design allow for the possibility of re-partitioning an application and reuse of modules. Where practical, bound your modules by business events and related units of work. The STA describes N-tier architecture in more detail – the architecture document is available at the IRMC website.

Programming management and supervisors must understand that the programming process for client/server applications is different from conventional mainframe-based development, primarily due to the languages, tools and platforms that are used. In particular, the distribution of effort within the programming process may be shifted; generators may simplify the coding process, but immature testing, debugging and configuration management tools may significantly increase the unit and string testing. Supervisors must have a complete understanding of how the process will work in their environment before embarking on this task.

VIII - UNIT TESTING

Unit Testing is the first step in a larger, detailed test plan (See testing phases below.) In addition to the work program, test conditions, and definitions of test cycles, the detailed test plan contains a bar chart showing all test cycles (logical groups of related test conditions), their scheduled start and stop dates and their inter-dependencies. This will clearly depict which cycles can be executed concurrently, and which must be run consecutively. The chart is updated to show actual dates as cycles are completed. This provides for easy monitoring of the system test and simplifies the process of rescheduling cycles and resources when required.

VIII.A Unit Testing Purpose

1) Test module logic.

VIII.B Unit Testing Objectives

1) Determine if the module meets the specifications.

2) Determine if the module functions without error.

VIII.C Unit Testing Output

Completing the steps outlined in Section VIII.D will result in a tested module and a Unit Testing Report. For each module, the report specifications are defined below:

1) Test Data - This section will document the test data and the test procedures used during unit testing.

2) Results - The results will document all the results from each test performed. This documentation will include defects in the module and differences between the module logic and the specifications.

The module specifications are defined below:

1) Specifications - The module will conform to the specifications.

2) Defects - The module is defect free.

VIII.D Steps to Complete the Unit Testing Report and Testing Module

Step	Description	Inputs	Outputs	Resources	Notes
1	Assign Modules to Testers		Module Assignments	Project Manager	Assignments Need to be Tracked. Steps 1- 8 Should be Repeated Until the Phase is Completed.
2	Test Module Against Specifications	Requirements Specification and Prototype Report	Differences between Specification and Tested Module	Tester	The Project Sponsor Should Be Involved as Often as Possible.
3	Create Test Data per Final System Report	Requirements Specification and Prototype Report – Final System Report	Test Data	Tester	
4	Test Module Against Test Data	Test Data	Defects Identified	Tester	
5	Verify Test Results	Results from Steps 2 & 4	Test Result Certification	Project Manager	
6	Document Test Results	Results from Steps 2, 3, 4 & 5 and Requirements Specification and Prototype Report	Module Testing Documentation	Tester	
7	Does Module Pass Testing?	Results from Steps 2, 4 & 6	If Module Passes, go to Step 9	Tester	
8	Initiate Change Request Element	Results from Steps 2, 4 & 6 and Change Request Element	Change Request	Tester	Steps 1- 8 Should be Repeated Until the Phase is Completed.
9	Finalize Unit Testing Report	Results from Step 6	Unit Testing Report	Project Manager	

IX - PHASE TESTING

IX.A Phase Testing Purpose

1) Test integration of individual modules into an overall phase.

IX.B Phase Testing Objectives

1) Determine if the phase meets the business requirements.

2) Determine if the phase meets the specifications.

3) Determine if the phase functions without error.

IX.C Phase Testing Output

Completing the steps outlined in Section IX.D will result in a tested phase and a Phase Testing Report and a phase ready for user acceptance. The report specifications are defined below:

1) Executive Summary - The executive summary must contain an overview of the phase testing procedures and results. Any significant defects or differences from the specification should be noted in this section because of the impact on schedule and resource requirements.

2) Test Data - This section will document the test data and the test procedures used during phase testing.

3) Results - This section will document all the results from each test performed. This documentation will include defects in the phase and differences between the phase logic and the specifications.

4) Signatures - The signature section of this document will be signed by the ITS Director and the project sponsor. A statement should precede the signatures identifying that the ITS Director and project sponsor accept the Phase Testing Report.

The phase specifications are defined below:

1) Business Requirements - The phase meets the business requirements outlined.

2) Specifications - The phase conforms to the defined specifications.

3) The phase will be defect free.

IX.D Steps to Complete Phase Testing Report and Testing Phase

Step	Description	Inputs	Outputs	Resources	Notes
1	Define Phase Testing Team		Team Roster	Project Manager	A Project Sponsor Should be Involved as Often as Possible.
2	Test Phase Against Business Requirements	Requirements Definition and Conceptual Design	Differences between Business Requirements and the Phase	Team	
3	Test Phase Against Specifications	Development Plan and Requirements Specification and Prototype Report	Differences between Specification and the Phase	Team	
4	Create Test	Requirements Specification and Prototype Report	Test Data	Team	
5	Test Phase Against Test Data	Test Data	Defects Identified	Team	
6	Verify Test Results	Results from Steps 2, 3 & 5	Test Result Certification	Team	
7	Does Phase Pass Testing?	Results from Steps 2, 3 & 5	If Phase Passes, go to Step 9; else go to 8	Team	
8	Initiate Change Request Element	Results from Steps 2, 3 & 5 and Change Request Element	Change Request	Team	
9	Prepare Phase Testing Report	Results from Step 2, 3, 4, 5 & 6	Phase Testing Report	Project Manager	
10	Obtain Signatures	Phase Testing Report	Signed Phase Testing Report	Team, ITS Director	

X - USER TESTING

X.A User Testing Purpose

1) To ensure that the phase deliverable meets the project sponsor's business requirements.

X.B User Testing Objectives

1) Determine if the phase meets the business requirements.

2) Determine if the phase meets the specifications.

3) Determine if the phase functions without error.

X.C User Testing Output

Completing the steps outlined in Section X.D will result in a phase that the project sponsor has accepted and a User Testing Report. The report specifications are defined below:

1) Executive Summary - The executive summary must contain an overview of the user testing procedures and results. Any significant defects and/or differences from the specifications and business requirements should be noted in this section.

2) Testing - This section will document the test data and the test procedures used during user testing.

3) Results - The results will document the outcomes of each test performed. This section will also document any defects and/or differences from the specifications and business requirements, .

4) Signatures - The signature section of this document will be signed by the ITS Director and the project sponsor. A statement should precede the signatures identifying that the ITS Director and project sponsor accept the Phase Testing Report.

The phase specifications are defined below:

1) Business Requirements - The phase will conform to the business requirements.

2) Specifications - The phase will conform to the specifications.

3) Defects - The phase will be defect free.

X.D Steps to Complete the User Testing Report and User Test the Phase

Step	Description	Inputs	Outputs	Resources	Notes
1	Define User Acceptance Team		Team Roster	Project Manager & Project Sponsor	Should Not Include Develop-ment Team
2	Create Test Data per the Acceptance Plan.	Requirements Specification and Prototype Report	Test Data	Team	
3	Test Phase Against Test/Live Data	Test Data/Live Data	Defects Identified	Team	
4	Test Phase Against Business Requirements	Requirements Definition and Conceptual Design	Differences between Business Requirements and the Phase	Team	
5	Test Phase Against Specifications	Requirements Specification and Prototype Report	Differences between Specification and the Phase	Team	
6	Verify Test Results	Results from Steps 3, 4 & 5	Test Result Certification	Team	
7	Does Phase Pass Testing?	Results from Steps 3, 4 & 5	If Phase Passes, go to Step 9; else go to Step 8	Project Manager	
8	Initiate Change Request Element	Results from Steps 3, 4 & 5 and Change Request Element	Change Request	Project Manager	
9	Prepare User Testing Report	Results from Steps 3, 4, 5 & 6	User Testing Report	Team	
10	Obtain Signatures	User Testing Report	Signed User Testing Report	Team, ITS Director, Project Sponsor	

XI – INTEGRATION TESTING

Client/server applications generally require more time and resources for system testing than traditional applications. There are several reasons: 1) The amount of time and resources required to perform the system test (specifically, the integration test) is largely a function of the number of system components to be integrated. Client/serve applications have many more components than traditional applications and many of these system components will reside on different computers (client workstations and servers); 2) Client/server applications often interface with other systems both within and outside the organization (e.g. Electronic Data Interchange), and these interfaces usually cannot be tested until the integration test; 3) There are more test conditions because there are more flexible user interfaces (which permit multiple ways for users to accomplish the same function); 4) It takes more effort to create the test data and expected results because processes and data are distributed among workstations and servers; 5) Stress tests are often required in these environments because of the overall complexity of the architectures and applications.

XI.A Integration Testing Purpose

Integration testing evaluates the integration of software subsystems (phases), hardware, and user support components, to validate the operational integrity of the application.

XI.B Integration Testing Objectives

1) To test the technical architecture components

2) To identify and address risk through integration testing of the technical architecture

3) To develop a testing approach that will serve as the model for the system test

4) Identify failures in the integration of system components.

XI.C Integration Testing Outputs

Completing the steps outlined in Section XI.D will result in an Integration Testing Report. The report specifications are defined below:

1) Executive Summary - The executive summary must contain an overview of the integration testing procedures and results. Any significant problems with integration should be noted in this section.

2) Integration - This section will include the test procedures used for testing the integration of the system components and result from the tests. Any problems should be noted.

3) Signatures - The signature section of this document will be signed by the ITS Director and the project sponsor. A statement should precede the signatures identifying that the ITS Director and project sponsor accept the Integration Testing Report.

The integration specifications are defined below:

1) Integration - The system successfully integrates all hardware and software components. Integration testing should test the interaction among components of the technical architecture, and between components of the technical architecture and the application. The goal is to ensure that different components will work properly once integrated.

XI.D Steps to Complete the Integration Testing Report and Test the System

Step	Description	Inputs	Outputs	Resources	Notes
1	Define the Integration Testing Team		Team Roster	Project Manager & Project Sponsor	Testers should be external to the project and as such have "new eyes".
2	Create Test Data	Requirements Specification and Prototype Report	Test Data that Tests System Against Integration Requirements	Team	
3	Create Test Scripts	Requirements Specification and Prototype Report; Technical Architecture	Test Scripts that Tests System Against Integration Requirements	Team	
4	Test Integration of System Phases	Phase Tested Modules	Tested Operational Configuration	Team	
5	Test Integration of Development, Execution, and Operational Architectures	Development, Execution, and Operational Architectures	Tested Technical Architecture	Team	See Reference XI.D.1
6	Test System Against Integration Requirements	1) Operational System Configuration 2) Tested Technical Architecture	Differences between Integration Requirements and the System	Team	See Reference XI.D.2
7	Verify Test Results	Results from Steps 4, 5 & 6	Test Result Certification	Team	
8	Does System Pass Testing?	Results from Steps 4, 5, 6, & 7	If System Passes, go to Step 11; Else go to Step 9	Project Manager	
9	Initiate Change Request Element	Results from Steps 4, 5, 6, 7 & 8 and Change Request Element	Change Request	Project Manager	
10	Quality Assurance Review	All Test Results and Procedures	Quality Assurance Comments	Quality Assurance Team	
11	Prepare Integration Testing Report	Results from Steps 4, 5, 6, 7 & 8	Systems Testing Report	Team	

Step	Description	Inputs	Outputs	Resources	Notes
12	Obtain Signatures	System Testing Report	Signed System Testing Report	Team, ITS Director, Project Sponsor	

References

XI.D.1 Test Integration of Development, Execution, and Operational Architectures

Prepare programming and testing standards for testers. Below are example criteria for setting standards.

Performance: Even if a benchmark is performed later, it is important to track performance characteristics and to fix performance problems during the integration test.

Reliability: As much as possible, test boundary conditions to ensure that the architecture components perform reliably and with integrity.

Usability: Ensure that architecture components are usable by developers and operators, and that they support the usability objectives of the application.

Flexibility: Ensure that the architecture isolates the effects of changes. Test specific portability and interoperability requirements.

Several interactions need to be tested:

Development architecture (Unified collection of tools and techniques used for constructing an application system): Ensure that different tools developed or acquired (perhaps from different vendors) will work properly once integrated. Be sure that any new tools or procedures added to the development architecture are included in the test. The architecture must be easy to use; a designer or programmer should not go through many complicated steps to develop a module.

Execution architecture (The run-time component of the production system, unified collection of run-time services, application infrastructure and controls upon which applications are constructed): It is especially important to test the interaction of components that cross platforms. A good example is a test cycle that requires a workstation to request data from a server. All aspects of the execution architecture (in particular, the data access services and distribution services) must work together to fulfill such a request.

Operations architecture (Combination of tools, support services, procedures, and controls to keep a system up and running. The primary users of this architecture are systems administrators and support staff.): This is an area where the project is likely to be using new or custom-developed tools. In an innovative environment, the operations architecture may have to address a set of issues that have not been encountered before. For example, a traditional mainframe shop may be faced with a new set of issues when downsizing to client/server applications running on multiple platforms (in particular, configuration management, data integrity and control, and system administration). It may not be possible to test the integration of operations tools with their corresponding procedures until later. In these cases, do the integration test of the operations architecture in stages. First test that the tools work together, then test the integration of the tools with their procedures.

Development and execution architectures: Test that components created or generated from the development architecture will function properly with the execution architecture. For example, a generated dialog should run successfully with the execution architecture. Also ensure that testing tools work properly with the execution architecture.

Development and operations architectures: There may be overlaps and interactions among components of these two architectures, in particular the configuration management components. Ensure that these areas work properly when integrated.

Execution and operations architectures: Focus on performance and resource monitoring and tuning, batch services and error handling. For example, test that performance monitoring hooks included in the execution architecture can properly feed the performance reporting components of the operations architecture.

Development, execution and operations architectures: Include test cycles that touch all areas of technical architecture, for example, data management. Use the development architecture to define and populate the test databases and to create the data manipulation statements. Use the execution architecture to process the data manipulation request, perhaps crossing platform boundaries. Use the operations architecture to synchronize multiple copies of data and to test backup/recovery scenarios for the databases.

The tools and procedures that are used here may provide a basis for the system test of the.

XI.D.2 Test System Against Integration Requirements

Test the interfaces (APIs) between the new application and the technical architecture components. Although these tests would usually include some script programs, ideally they use actual application code being developed.

Finally, testing must confirm that new or revised software does not have an unacceptable impact on the performance of existing software. These tests are especially crucial for Shared Services that are used across the enterprise.

It is also important to verify that the architectures being developed are compatible with other non-application related software typically found on a user's workstation, for example, word processors, spreadsheets and electronic mail.

XII - SYSTEM TESTING

XII.A System Testing Purpose

System testing qualifies the delivered system against the design requirements. System design is verified, system requirements are validated, and operational integrity is established.

XII.B System Testing Objectives

1) Identify failures in the ability of the system to conform to business requirements.

XII.C System Testing Outputs

Completing the steps outlined in Section XII.D will result in a System Testing Report. The report specifications are defined below:

1) Executive Summary - The executive summary must contain an overview of the system testing procedures and results. Any significant differences from the business requirements should be noted in this section.

2) Business Requirements - This section will contain the test procedures used for testing the business requirements and results from the tests. Any differences, no matter how small, should be noted in this section.

3) Signatures - The signature section of this document will be signed by the ITS Director and the project sponsor. A statement should precede the signatures identifying that the ITS Director and project sponsor accept the System Testing Report.

The system specifications are defined below:

1) Business Requirements - The system meets the specified business requirements.

XII.D Steps to Complete the System Testing Report and Test the System

Step	Description	Inputs	Outputs	Resources	Notes
1	Define the System Testing Team		Team Roster	Project Manager & Project Sponsor	Testers should be clients who have participated in the design of and/or will use the system.
3	Create Test Data	Operational System Configuration, Requirements Specification and Prototype Report	Test Data that Tests System Against Business Requirements	Team	
4	Create Test Scripts	Operational System Configuration, Requirements Specification and Prototype Report	Test Scripts that Tests System Against Business Requirements	Team	
5	Test System	Test Scripts/Data	Defects Identified	Team	
6	Verify Test Results	Results from Steps 4	Test Result Certification	Team	
7	Does System Pass Testing?	Results from Steps 4	If System Passes, go to Step 8; Else to Step 7	Project Manager	
8	Initiate Change Request Element	Results from Steps 4, 5, 6 and Change Request Element	Change Request	Project Manager	
9	Quality Assurance Review	All Test Results and Procedures	Quality Assurance Comments	Quality Assurance Team	
10	Prepare Systems Testing Report	Results from Steps 4, 5, 6, 7 & 8	Systems Testing Report	Team	
11	Obtain Signatures	System Testing Report	Signed System Testing Report	Team, ITS Director, Project Sponsor	

XIII - IMPLEMENTATION

XIII.A Implementation Purpose

To develop and execute an action plan for the implementation of the project.

XIII.B Implementation Objectives

1) Develop and implement a training plan, which outlines the training requirements for the users and/or MIS staff.

2) Develop and execute system implementation procedures for placement of the project/phase into production.

XIII.C Implementation Output

Completing the steps outlined in Section XIII.D will result in a successful implementation of the project/phase and an Implementation Report. The report specifications are defined below:

1) Executive Summary - The executive summary will contain the beginning and ending dates for the implementation process, a summary of the action plan, training requirements, and staffing requirements.

2) Gantt Chart - The Gantt Chart will identify the time-line for each action to be taken to implement the project.

3) Implementation Plan - The implementation plan must include a description of the actions to implement the project/phase. This description should include both funding and personnel resources required for each action and the time frame to complete the action.

4) Training Plan - This section will include the training required for both project sponsor and MIS staffs. In addition, the training plan should include actions to accomplish the training, resources required and the time frame to complete each action.

5) Signatures - The signature section of the document will be signed by the ITS Director and the project sponsor. A statement should precede the signatures identifying that the ITS Director and project sponsor accept the Implementation Plan Report.

XIII.D Steps to Complete the Implementation Report, Perform Training and Implement the Project/Phase

Purpose: Develop and implement a training plan which outlines the training requirements for the project sponsor and/or MIS staff.

Step	Description	Inputs	Outputs	Resources	Notes
1	Establish the Implementation Training Team		Team Roster	Project Manager & project sponsor	At least two people, 1 MIS and 1 Project Sponsor
2	Develop a Detailed Training Plan for All Affected Staff	Knowledge of Current/Required Skills	Detailed Training Plan	Team	
3	Identify Personnel to Receive Training	Results from Step 2	List of Trainees	Team	
4	Select Instructor(s)		Instructor(s)	Team	
5	Determine Technique	1) Knowledge of Current/Required Skills 2) Instructor Knowledge	Training Techniques	Team, Instructor(s)	See Reference XII.D.1
6	Select/Develop Training Material	1) Knowledge of Current/Required Skills 2) Knowledge of Training Techniques	Training Materials	Team, Instructor(s)	
7	Schedule and Conduct Training	Instructor(s), Implementation, and Resource Schedules	Training Complete	Team, Instructor(s)	
8	Post-Training Follow-up	Trainee Feedback	Revised Training Plan	Team, Instructor(s)	

References

XIII.D.1 Determine Technique

Some general training methods are more appropriate for some groups of users than others. A complete training program for any user group combines several methods. The training methods may include centralized workshops and seminars, classroom instruction, case studies and self-directed training.

Seminars and small group meetings

Seminars and small group meetings promote extensive two-way communication. Particularly effective for training management personnel, the meetings are also an excellent way to present systems concepts, to demonstrate how to use information produced by the system in making management decisions, and to gain management acceptance of the system.

Classroom instruction

This method is more appropriate for training non-management users in the daily use of the system output and the procedural changes in their operating environment. Classroom instruction should include discussion periods to explain specific points. During these sessions, the users should become familiar with the documentation and reference aids.

Classroom exercises

All training sessions should include exercises and case studies in areas that require a thorough understanding of the detailed requirements, familiarity with the user interface or proficiency in performing repetitive operations.

Exercises stimulate interest in the training session by giving the trainees a sense of achievement from the training. Conduct exercises on the actual system equipment.

Independent study

Independent study may be appropriate for decentralized system users, user groups that are only slightly affected by the new system, or during introductory training for major training sessions. The activities contained in independent study units can be self-taught or taught with minimal instructor assistance. Computer-based training (CBT) is an obvious candidate for learning about a computer-based system.

There should be a follow-up discussion with the participants to answer questions and ensure that they achieve training objectives.

Programmed instruction

Programmed instruction courses can be effective in certain areas if they are very well prepared. This requires either experienced personnel or professional services. CBT is one form of programmed instruction that is especially well adapted to training users of computer applications. Generally, CBT is a complex and high-cost solution, often warranting a development project of its own.

Guided instruction

Guided instruction combines independent study and instructor-led training. The instructor introduces the topic, answers questions about the material, and explains the relationship of each topic to the complete systems operation. The instructor may also lead a general discussion. The bulk of the training is independent study.

Purpose: Develop and execute system implementation procedures for placement of the project into production.

Step	Description	Inputs	Outputs	Resources	Notes
1	Establish the Implementation Team		Team Roster	Project Manager & Project Sponsor	
2	Determine Actions Required to Implement the Project	1) Requirements Definition and Conceptual Design Report 2) Architecture Report	Implementation Action Plan	Team	See reference XIII.D.2, XIII.D.3, XIII.D.4
3	Determine Person-hours	Results from Step 2	Person-hours Required	Team	
4	Determine Staffing Requirements and Skills Needed	Results from Steps 2 & 3	Additional Staffing	Team	
5	Determine Materials Needed	Results from Steps 2 & 4	Materials Required	Team	
6	Create Gantt Chart	Results from Steps 2, 4 & 5	Gantt Chart	Team	
7	Prepare Implementation Report	1) Results from Previous Steps 2) Training Plan	Implementation Report	Team	
8	Obtain Signatures	Implementation Report	Signed Implementation Report	Team, ITS Director, Project Sponsor	
9	Implement Project/Phase	Results from Step 8	Implemented Project/Phase	Team, Additional Staffing	
10	Test Project/Phase	Results from Step 9	Required Implementation Modifications	Team, Additional Staffing	Steps 10-11 Should be Repeated Until the Project/ Phase Functions Properly.

Step	Description	Inputs	Outputs	Resources	Notes
11	Implement Modifications	Results from Step 10	Modified Project/Phase	Team, Additional Staffing	Steps 10-11 Should be Repeated Until the Project/ Phase Functions Properly.

References

XII.D.2 Actions Required to Implement Project

The following items should be considered when drafting an implementation action plan: 1) scheduling with existing operations personnel, 2) spelling out a detailed contingency plan should severe problems occur, 3) provisions for backup and file conversion and "data scrubbing" (including adequate control to assure conversion files are properly maintained until they are input to the new system), 4) fully spelled-out system test plan (including provisions for such things as on-line transaction processing, documentation to measure the performance of the system, selection of "real world" test data and performance requirements, 5) a provision to test the system more than once, 6) archival of files, procedures and other documents that are being replaced, 7) appropriate user participation and signoff in the development of the implementation action plan, 8) a change management procedure for controlling problems, and 9) written conversion procedures.

A key set of issues to manage during implementation relates to the technology architecture. Have all updates to the technology architecture been documented? Has the technology configuration been tested to establish performance/capacity? Have components that were prone to problems been fixed7 Has IT been involved in resolving/understanding any technology issues?

For each new or modified database, identify sources of data, identify integrity constraints, identify co-existence and interface requirements to other systems. Design overall data conversion flow including:

- Validation
- Cleanup
- Loading
- Maintaining before going live

Design data conversion programs, and plan for data conversion contingencies

The implementation must also include the design and initiation of a maintenance environment for support of the production system. Special consideration must be given in planning the deployment of Shared Services that the project has developed (deployment requirements must be discussed with the NCSB support group) – key items include, subscription fees, subscription process, change management, and archiving of test drivers and procedures (that may be reused by future service users or for regression testing).

XII.D.3 Actions Required to Implement Project - Documentation

Below are considerations to keep in mind when developing user and system documentation.

Shared Services

A standard form of documentation is required for shared services using the NC Service Broker. This includes the request message parameters and response message layout, as well as the business and

processing rules performed by the shared service. This documentation becomes part of the Statewide or Departmentwide component repository. The current shared service documentation requirements is included in Section 7 – Documentation Guide. For the most updated version, contact the NCSB Support Group.

User procedures

User procedures are those required to operate the user interface, or to prepare input to or use output from the system. The user procedures are written in user language. User procedures are frequently grouped into the following categories:

Workstation operation

Workstation operation procedures document the log-on/log-off procedure, the various dialog flows, and the security and control functions associated with the online system. Help keys, menu-driven systems, and windows containing field descriptions and processing facilities can incorporate these procedures into the online system design.

Use and transmission of forms

Form-filling and transmission procedures document all steps necessary to create and process forms.

Computer systems report distribution

Computer systems report distribution and usage procedures document the distribution of the output and describe how to interpret and use it. There may be information on how to use the system outputs for several levels of users, from clerical personnel to departmental supervisors and management personnel.

Error and suspense correction and reentry

Error and suspense correction and reentry procedures specify who is responsible for investigating and completing the identification, correction and reentry procedures.

Basic administrative/system documentation may include the categories below.

Organization, supervision, and work flow controls

Organization, supervision and work control procedures address activities that require many personnel to perform a single task.

Security

Security procedures serve two purposes: They prevent unauthorized modification, disclosure, and destruction of information and other systems resources; and they provide fallback and recovery processing in case of destruction or disaster. The latter procedures should address short-term and long-term contingencies. If the organization has a contingency plan, these procedures should be integrated into the plan. The specific risks that security procedures address were identified during Quality Requirements.

Control procedures

Control procedures verify the accuracy and reliability of the system and the integrity of the data. Control functions can be performed by a separate production control group within the operations area or by the various user departments. The project team must define specific control responsibilities for each group. Controls provide the audit trails necessary to follow a transaction from its originating authorization through processing to the end product. These controls usually combine manual and computerized activities.

The project team documented the necessary controls in the Quality Requirements segment. The user interface design and the technical design incorporate these requirements. In this task, check their completeness and update the designs as required.

Overall systems controls

Overall systems controls ensure that the controls for each operation in a system are accurate and reliable. They ensure that data to be processed is entered into the system and is processed correctly through the manual and automated steps, and that the overall results are accurate, complete and reported. These controls are a quick, overall check of the integrity of the entire system, especially the master files. They may also provide physical inventory procedures that periodically verify the data maintained on master files.

Input controls

Input controls ensure that all transactions that should be processed actually are. Transactions are logged at the earliest possible moment and are reconciled as they are entered into the computer system. These procedures also include the correction and reentry of erroneous transactions that are in suspense.

Programmed controls

Programmed controls are designed by the analysts and executed by the computer. They ensure that data entered into the system is valid, correctly passed between programs, posted to the proper files, and reported. Although these controls are largely automated, manual procedures are required to make sure that out-of-balance conditions are discovered and corrected.

Output controls

Output controls ensure that reported information is complete and accurate, and that it is prepared and distributed as intended. These procedures include specific steps for reviewing and balancing output reports; verifying that all required reports are prepared; verifying that pre-numbered and computer-numbered forms are properly used; and verifying that all output is properly distributed.

Physical and administrative controls

Physical and administrative controls ensure the proper performance of computer functions such as physical control of computer programs, data files and operators. These procedures relate primarily to the overall management and control of IS operations. The project team should review and update the existing procedures as necessary before implementing the new system.

Work flow control

How the work flow is controlled depends to a large extent on the style of the application and the level of the users.

In case-processing systems operated by knowledge workers, controlling the work flow may include performance monitoring, case status tracking, and exception notification and approval:

Performance monitoring consists of setting objectives for each worker and measuring whether they have been attained. Performance monitoring should be designed by organizational change experts working closely with the organization's human resources department. If performance monitoring is to be automated, iterate through the design cycle to incorporate the required functionality in the automated process flow.

Tracking case status consists of reporting work in process, what remains to be done, and possibly the time frame in which it is to be done. This functionality is easy to automate and must be included in the automated process design.

Notification and approval of exceptions is often online, providing supervisors with the authority to review completed work and to dispose of open points when the case worker does not have authority to make a decision. Add the corresponding dialogs with the required functionality to the user interface design.

In traditional transaction-processing systems operated by clerical personnel with little or no decision authority, work flow control involves logging all work as it reaches the worker, and then reconciling the output to the input to ensure that all submitted work has actually been done. If supervisory approval is needed, it is usually documented on a hard copy of the relevant document. Performance monitoring is usually by direct observation. Work in process is usually insignificant over the long run, since each transaction is completely processed as it arrives; however, it may be necessary to control the contents and aging of suspense files or accounts.

XIV PRODUCTION SYSTEM SUPPORT

XIV. A Production System Support Purpose

1) To ensure that production systems operate efficiently and effectively modified with minimum disruption.

XIV.B Production System Support Objectives

1) To provide error-free, efficient operation

2) To respond to a changing business environment

3) To monitor the performance of the production system

4) To document system modifications or enhancements

5) To initiate the implementation of a modification or enhancement

XIV.C Production System Support Output

Completing the steps outlined in Section XIV.D will result in successful production support resources.

1) Identification of a Client Representative.

2) Periodic System performance reports

3) Periodic change requests, log of change requests, and change status reports

4) Periodic system modification plans and modifications

5) Approved change acceptance checklists. A statement should precede the signatures identifying that the ITS director officer and the user liaison accept the modifications

XIV.D Production Support Steps

Step	Description	Inputs	Outputs	Resources	Notes
1	Select a representative to identify, initiate and approve changes		Client Representative	Senior manager (e.g. director) of sponsoring organization	
2	Organize and staff production system support				See Reference XIV.D.1
3	Develop Monitoring Procedure and Review Schedule	Technical Architecture, Operations Management Procedures, Detailed Design Specifications	Review Schedule	Project Manager	
4	Conduct Annual Reviews	Technical Architecture, Operations Management Procedures, Detailed Design Specifications, Review Schedule	Change Requests (where justified) and Change Request Priorities	Representative, assigned ITS personnel	
5	Batch Change Requests	Change Request Priorities	Group Change Requests	Representative, assigned ITS personnel	See Reference XIV.D.2
6	Implement and Manage Change Requests				See Change Management Phase XVI
7	Conduct Acceptance Tests and obtain sign-off	New work units	Approved work units	Designated client users and production support staff	See Reference XIV.D.3

References

XIV.D.1 Organize and Staff Production System Management

Organizing for production systems support consists primarily of allocating resources to major production systems areas and assigning priorities to change requests. Initial staffing allocations should be based on the business needs of the department and on budgetary restrictions. Resource allocations should be flexible to allow production systems management to respond to unexpected demands, particularly mandatory changes. The level of support required for a system will vary over the life of the system. Immediately following conversion, support activities will be quite high to correct errors and process change requests identified curing the systems installation. The initial maintenance team for the system should include personnel who participated in the system's development. Responsibility for the maintenance of specific applications should rotate among maintenance personnel. This ensures that the continued operation of a system is not dependent on one individual. However, the rotation should not be so frequent that it disrupts

the orderly maintenance of systems. A transition in responsibilities should include a staffing overlap as experienced personnel are phased out of an application and inexperienced are phased in.

XIV.D.2 Batching Change Requests

The SDT Chief should establish a policy and guidelines for version control in the maintenance and release of updates to systems. Version control is grouping or batching individual change requests in one systems area for simultaneous processing and implementing. Version control reduces the effort required to process charge requests and minimizes disruptions while implementing the changes. It will also facilitate control over the production support activities by working in fewer and larger projects rather than by individual change requests. An example of specific policy for version control in supporting systems could be the release of a major update or new version of the system every 12 months. Each new version could be numbered sequentially, with change requests filed by the version in which they were released. Interim updates to correct system errors, for example, would be numbered in a manner that would, directly reference the version they are updating The client representative, ITS Director, and SDT Chief must also determine which change requests will be released in each system update (or version) and plan the release of the updates.

XIII.D.3 Acceptance Tests

Acceptance tests include both a test by the end user to review the functionality of the new work units, and they should include integration tests to assure that the interface of the new work units with existing units and components is satisfactory.

XV - PROCESS ASSESSMENT

XV.A Process Assessment Purpose

1) To review Change Management Report(s) generated during the current project in an attempt to avoid future alterations by providing recommendations for improving the Departmental system development life cycle methodology (SDLC).

XV.B Process Assessment Objectives

1) Define recommendations for improving the Departmental SDLC.

XV.C Process Assessment Output

Completing the steps outlined in Section XV.D will result in a Process Assessment Report that will outline recommendations for improving the SDLC. The report specifications are defined below:

1) Analysis - The analysis will outline the method followed for reviewing the Change Management Report(s) in the brainstorming session(s).

2) Identification - This section will identify the element in the SDLC where the error in each Change Management Report was introduced. A table summarizing the total for each element should be included.

3) Common Themes - This section will contain common themes identified in the analysis and any relationships established.

4) Recommendations - The recommendations will outline potential modifications to the SDLC that will help avoid alterations in the future and thus reduce the cost of system development. This section must outline the recommended change, how the recommended change will improve the SDLC, and the related analysis.

5) Signatures - The signatures section of this document will be signed by the ITS Director. A statement should precede the signatures identifying that the ITS Director accepts the Process Assessment Report.

XV.D Steps to Complete the Process Assessment Report

Step	Description	Inputs	Outputs	Resources	Notes
1	Define the Process Assessment Team	Team Rosters from All Previous Elements	Team Roster	Project Manager, Project Sponsor	At least two people, 1 MIS and 1 Project Sponsor
2	Review All Change Management Reports	All Change Management Reports		Team	
3	Determine Analysis Process		Analysis Method and Outline for Brainstorming Session(s)	Project Manager	Distribute at Least Two Days Prior to First Brainstorming Session
4	Brainstorming Session(s)	1) Change Management Reports 2) Knowledge of Process Completed	1) Error Introduction Point 2) Common Themes 3) Recommendations and Justifications	Team	This May Take Multiple Sessions
5	Prepare Process Assessment Report	Results from Steps 3 & 4	Process Assessment Report	Team	
6	Obtain Signatures	Process Assessment Report	Signed Process Assessment Report	Team, ITS Director	
7	Modify the SDLC	Signed Process Assessment Report	Modified SDLC	Team	

XVI - CHANGE MANAGEMENT

XVI.A Change Management Purpose

1) To assess the impact (schedule, cost, function, etc.) of a potential change and implement the change. A change is a modification of any specification after it has been approved or a defect created during implementation (also know as an alteration). Changes are assessed by going through the entire project planning process with the change included and then comparing the updated project plan with the original. The difference between the two project plans is the impact of the change.

XVI.B Change Management Objectives

1) Identify when and why a change occurs.

2) Assess the potential change.

3) Develop a process for integrating the change.

4) Obtain approvals to implement alteration.

XVI.C Change Management Output

Completing the steps outlined in Section XVI.D will result in a Change Management Report, potential modifications of other documents and potential implementation of a project modification. The report specifications are defined below:

1) Change Request - The change request will document the desired alteration. This section will include a reference to the element, section and page of the specification that is being changed and the reason for the change.

2) Change Analysis - The change assessment must document the effect the alteration will have on the project schedule, cost, person-hours, and staffing. This section will also include the elements in which the change is being identified.

3) Change Process - The change process will include the actions to be taken to implement the alteration. This section will also include the resources and time required for each action.

4) Signatures - The signatures section of this document will be signed by the ITS Director and the project sponsor. A statement should precede the signatures identifying that the ITS Director and project sponsor accept the Change Management Report and the commitment of additional resources.

Reports modified because of the alteration of a specification:

1) Alteration - All reports affected by the alteration must be updated to reflect the change approved.

2) Signatures - The signatures section of the document(s) will be signed by the ITS Director and the project sponsor. A statement should precede the signatures identifying that the ITS Director and project sponsor accept the revised documents.

XVI.D Steps to Complete the Change Management Report and Update Changed Reports

Step	Description	Inputs	Outputs	Resources	Notes
1	Prepare Change Request		Change Request	Project Sponsor/Client Representative	Only Project Sponsor (during the development phase) or Client Representative (when application is in production) can submit change request
2	Prepare Change Analysis	Change Request	Impact of Change Request	Project Manager/Team during development; STD Chief during production period	See reference XVI.D.1
3	Prepare Draft Change Management Plan	Results from Steps 1 & 2	Draft Change Management Report	Project Manager/Team during development; STD Chief during production period	
4	Obtain Preliminary Approval and Commitment	Draft Change Management Plan	Signed Change Management Plan	ITS Director, Project Sponsor	
5	Implement Change	Signed Change Management Plan	Implemented Change	Project Manager/Team during development; STD Chief during production period	See reference XII.D.2
6	Update Change Management Status Report	Change Management Status Report	Updated Change Management Report	Project Manager/Team during development; STD Chief during production period	See reference XVI.D.2

References

XVI.D.1 Prepare Change Analysis

The change analysis task expands the change request requirements into a completed change design and implementation plan. These expansion activities typically consist of the following:

1) Development specifications for minor program changes

2) Conceptual design for new programs and major changes to existing programs (This may require extensive design activities and would benefit from referencing earlier phases of the SDLC.)

3) Identification of revisions

4) Definition of a testing approach development of a detailed work program

5) Preparation of a cost and benefit analysis

XVI.D.2 Update Change Management Status Reports

A status report should be updated for each application and for the department's application portfolio. These reports should include such status categories as resolved, in-process, and backlogged change requests.

End of Section 4